# FORE Systems PowerHub Supplementary Protocols Manual, Software Version 2.6

PN 400-1676-0001, Rev C, Issue 1, July 1996

FORE Systems reserves the right to revise this publication and to make changes in its content without obligation on the part of FORE Systems to provide notification of such revision or change. FORE Systems provides this manual without warranty of any kind, either implied or expressed, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. FORE Systems may make improvements or changes in the product(s) and/or the programs described in this manual at any time.

**FORE Systems**
174 Thorn Hill Road
Warrendale, PA 15086-7586
Phone: 412-772-6600
Fax: 412-772-6500
**Product Information:**
Phone: 1-888-404-0444
Fax: 412-635-6325
**Technical Support:**
1-800-671-FORE
Internet: info@fore.com
or http://www.fore.com

## PUBLICATION HISTORY

PowerHub Supplementary Protocols Manual, Software Version 2.6

PN 400-1676-0001

| Date | Rev | Issue | Description |
|---|---|---|---|
| June, 1995 | A | 1 | First edition, corresponding to system software version 2.6. |
| August, 1995 | B | 1 | Second edition, corresponding to system software version 2.6. |
| July, 1996 | C | 1 | Third edition, corresponding to system software versions 7-2.6.3.0, 6-2.6.3.0, and 4-2.6.1.2. |

## ABOUT THIS BOOK

This book is for anyone using FORE Systems PowerHub software version 2.6. This book is divided into the following parts:

| | |
|---|---|
| Part 1: Subsystems | Describes the commands you can use to configure the hub for AppleTalk, DECnet, and IPX routing, as well as information on implementing IP security. |
| Part 2: Filtering | Describes the commands you can use to configure AppleTalk zone and forwarding filters, and IPX RIP and SAP filters. |

**NOTE**: This manual and the *PowerHub Software Manual, V 2.6* (Rev C) describe the current system software version as “2.6.” However, when you receive software diskettes containing PowerHub software files, the diskettes are labeled according to the specific PowerHub product and software release. For example, a software diskette shipped for a PowerHub 7000 might be labeled “7-2.6.3.0.” The “7-” indicates that the software is for the PowerHub 7000 and the “3.0” indicates the specific feature or maintenance release.

# OTHER BOOKS

You can find additional information about the PowerHub system in the following books:

*PowerHub Software Manual, V 2.6* (Rev C)
Describes the commands to configure the PowerHub 7000 for bridging and for IP routing.

*PowerHub ATM Addendum* (Rev A or higher)
Describes the PowerCell ATM modules, as well as how to install them and how to configure them for LANE (LAN Emulation) 1.0.

*PowerHub OSPF Addendum* (Rev A or higher)
Describes the PowerHub implementation of the OSPF (Open Shortest Path First) routing protocol and how to configure your PowerHub system as an OSPF router.

*PowerHub 4000 Installation and Configuration Manual, V 4-2.6.1.2* (Rev B)
Describes the PowerHub 4000 hardware, as well as how to install it and how to upgrade it.

*PowerHub 4005 FDDI Addendum* (Rev A or higher)
Describes the PowerHub 4005 FDDI hardware and how to install it and configure it.

*PowerHub 4100 Fast Ethernet Addendum* (Rev A or higher)
Describes the PowerHub 4100 hardware and how to install it and configure it.

*PowerHub 6000 Installation and Configuration Manual, V 6-2.6.3.0* (Rev C)
Describes the PowerHub 6000 hardware, as well as how to install it and how to upgrade it.

*PowerHub 6000 FDDI Addendum* (Rev A or higher)
Describes the PowerHub 6000 FDDI daughter cards and how to install them and configure them.

*PowerHub 7000 Installation and Configuration Manual, V 7-2.6.3.0* (Rev D)
Describes the PowerHub 7000 hardware, as well as how to install it and how to upgrade it.

*PowerHub 7000 6x1 Universal Fast Ethernet Addendum* (Rev A or higher)
Describes the PowerHub 7000 6x1 Fast Ethernet Module, as well as how to install it and how to upgrade it.

## NAMING CONVENTIONS

The following terms are used throughout this manual:

*Port* A physical connection to the PowerHub system. A given segment can have many ports; for example, repeated ports on a UTP Ethernet segment. A segment also can have just one connection; for example, an AUI segment, in which case the terms “port” and “segment” are somewhat interchangeable.

*Segment* A single 10 Mb/s or 100 Mb/s Ethernet collision domain or a 100 Mb/s FDDI ring.

**NOTE**: To conform with industry-standard usage of terms, FORE Systems now uses the words “segment” and “port” as they are defined in the preceding list. This usage is reflected in PowerHub documentation, including this manual.

Previous FORE Systems products and documentation used the word “port” to refer to a single collision domain or ring, and “link” to refer to a physical connection. In PowerHub software versions 2.6 and earlier, commands and help messages in the command-line interface continue to be based on the old usage.

# TYPOGRAPHICAL CONVENTIONS

The following typographical conventions are used in this manual:

**This type style...** **Indicates...**

*AaBbCcDd* A term that is being defined. Example:

> *IP Helper* is an enhancement to the **ip** subsystem that lets you boot a PowerHub system from a server separated from the boot client by a gateway.

**`AaBbCcDd`** A command name. PowerHub commands are case sensitive; they should always be entered as shown in the manual or on-line help. Example:

```
listdir
```

**`|`** 1) Separates the full and terse forms of a command or argument:

- The full form is shown on the left of the **`|`**.
- The terse form is shown on the right of the **`|`**.

Example:

```
listdir|ls
```

When you type the command or argument, you can type either the full form or the terse form. In this example, you can type **`listdir`** or **`ls`**.

2) Separates mutually exclusive command arguments. Example:

```
set stp enl|dis
```

In this example, the command **`set stp`** can accept either **`enl`** or **`dis`**, but not both.

**`[ ]`** Enclose optional command arguments or options. Example:

```
setuser [root|monitor]
```

In this example, the **`[ ]`** enclose an optional argument. You can issue the command without the argument(s) shown in **`[ ]`**. However, if specified, the argument must be one of the two options listed between the **`[ ]`**.

<*`AaBbCcDd`*> Indicates a parameter for which you supply a value. When used in command syntax, <*`italics`*> indicates a value you supply. Example:

```
showfile <file-name>
```

In this example, <*`file-name`*> is a parameter for which you must supply a value when you issue this command.

`AaBbCcDd` Indicates a field name or a file name.

A field name example:

When you boot the PowerHub software, the `login:` prompt is displayed.

A file name example:

When you boot the PowerHub software, the system looks for a file named `cfg`.

```
AaBbCcDd
or
AaBbCcDd
```

Indicates text (commands) displayed by the PowerHub software or typed at the command prompt. To make typed input easy to distinguish from command prompts and output, the typed input is shown in darker type. Example:

```
1:PowerHub# pm view all 10 on 12
Port 10 (all) being viewed on: 12
```

In this example, the user types “**`pm view all 10 on 12`**” and the software responds “`Port 10 (all) being viewed on: 12`”.

# Contents

## Part 1: Subsystems

# Part 2: Filtering

# Part 1: Subsystems

This part describes the PowerHub subsystem commands in the **atalk**, **ipx**, and **dec** subsystems, as well as the commands used to configure IP security.

- If you want information on the **bridge**, **tcpstack**, **tftp**, **ip**, **ipm**, **rip**, or **snmp** subsystems, see the *PowerHub Software Manual, V 2.6* (Rev C).
- If you want information about IPX RIP and SAP filtering, go to Part 2: Filtering.

This part contains the following chapters:

# 1 AppleTalk Commands

The PowerHub software contains the **atalk** (AppleTalk) subsystem. Within this subsystem is a complete set of AppleTalk Phase-2 software and PowerHub software for use with AppleTalk networks and internets.

You can configure the PowerHub system to be used as an AppleTalk internet router to perform AppleTalk routing on any or all of its segments. You also can use the PowerHub system as a local router or a backbone router, or as any combination of these types of routers.

This chapter describes the **atalk** subsystem commands you can use to perform the following tasks:

- Allocate memory for the AppleTalk subsystem. (See Section 1.3.1 on page 7.)
- Enable AppleTalk routing. (See Section 1.3.1.2 on page 8.)
- Show the current AppleTalk configuration. (See Section 1.3.1.3 on page 8.)
- Add an AppleTalk interface. (See Section 1.5.1 on page 12.)
- Delete an AppleTalk interface. (See Section 1.5.4 on page 18.)
- Display the AppleTalk interface table. (See Section 1.5.2 on page 14.)
- Display the AppleTalk route table. (See Section 1.7 on page 22.)
- Display and clear the AppleTalk route cache. (See Section 1.8 on page 24.)
- Add an AppleTalk zone. (See Section 1.4.1.1 on page 9.)
- Display a list of configured AppleTalk zones. (See Section 1.4.1.2 on page 10.)
- Display a list of active (static and learned) AppleTalk zones. (See Section 1.4.1.2 on page 10.)
- Delete an AppleTalk zone. (See Section 1.4.1.3 on page 11.)
- Set the aging time for entries in the AppleTalk ARP table. (See Section 1.6.2 on page 21.)

- Display the AppleTalk ARP table. (See Section 1.6.1 on page 20.)
- Clear the AppleTalk ARP table. (See Section 1.6.3 on page 22.)
- Display a table of "named objects." (See Section 1.9 on page 26.)
- Display statistics for AARP, DDP, or ECHO packets. (See Section 1.10 on page 26.)
- Clear packet statistics. (See Section 1.11 on page 28.)
- Test the connectivity to another router. (See Section 1.12 on page 28.)

This chapter assumes that you have a basic understanding of AppleTalk. For further information about AppleTalk, see the following books:

- *Inside AppleTalk, 2nd Ed.*, by Gursharan S. Sidhu, Alan B. Andrews, and Richard F. Oppenheimer (Addison Wesley Publishing Company, Inc., 1990).
- *Planning and Managing AppleTalk Networks*, by Apple Computer (Addison Wesley Publishing Company, Inc., 1991).

## 1.1 ACCESSING THE APPLETALK SUBSYSTEM

To access the **atalk** subsystem, enter the following command from the runtime command prompt:

```
atalk
```

Most of the commands in this chapter assume that you have changed the focus of the command prompt to "`atalk`." A few commands, such as **`getmem`**, are not in the **`atalk`** subsystem. This chapter identifies such commands by listing their subsystem name with the command (ex: **`main getmem`**.)

## 1.2 APPLETALK SUBSYSTEM COMMANDS

Table 1–1 lists and describes the **atalk** subsystem commands and their syntax. For each command, the management capability (root or monitor) is listed, as well as the section that contains additional information about the command.

**TABLE 1–1** AppleTalk subsystem commands.

| Command and Description | Capability* | See… |
|---|---|---|
| **add-interface\|ai** *<seg-list>* *<start-net-addr>***-***<end-net-addr>***\|-n** *<net-addr>***.***<node-addr>* **[***<default-zone-name>***]**<br>Adds an AppleTalk interface. | R | 1.5.1 |
| **add-zone\|az** *<seg-list>***\|all** *<zone-name>*<br>Adds an AppleTalk zone. | R | 1.4.1.1 |
| **arp-table\|at [***<net-addr>***.***<node-addr>***]**<br>Displays the AppleTalk ARP table. | R or M | 1.6.1 |
| **arp-tableclear\|atc**<br>Clears the AppleTalk ARP table. | R | 1.6.3 |
| **config-zone-table\|czt [***<seg-list>***\|all]**<br>Displays a list of configured AppleTalk zones. | R or M | 1.4.1.2 |
| **configured-interface-table\|cit**<br>Displays information about all the AppleTalk interfaces configured on the PowerHub system. | R or M | 1.5.3 |
| **del-interface\|di [-a]** *<seg-list>***\|all**<br>Deletes an AppleTalk interface. | R | 1.5.4 |
| **del-zone\|dz** *<seg-list>***\|all** *<zone-name>***\|all**<br>Deletes an AppleTalk zone. | R | 1.4.1.3 |
| **display-routecache\|dc [***<seg-list>***]**<br>Shows the Appletalk route cache. | R or M | 1.8.1 |
| **flush-routecache\|fc**<br>Flushes (clears) the Appletalk route cache. | R | 1.8.2 |
| *R= Root, M=Monitor. | | |

**TABLE 1–1** (Continued) AppleTalk subsystem commands.

| Command and Description | Capability* | See… |
|---|---|---|
| **interface-table\|it**<br>[**-p** *<seg-list>*\|**all**]<br>\|[**-n** *<start-net-addr>*-*<end-net-addr>*]<br>\|[**-z** *<zone-name>*]<br>\|[**-a**]<br>Displays the AppleTalk interface table. | R or M | 1.5.2 |
| **name-table\|nt**<br>Displays the table of “named objects” (objects named using the AppleTalk Name-binding protocol). | R or M | 1.9 |
| **ping\|pi**<br>*<net-addr>*.*<node-addr>* [*<time-out>* [*<pktsize>*]]<br>Tests connectivity to another router. | R or M | 1.12 |
| **route-table\|rt**<br>**[-c\|-r] [-t] [-p** *<seg-list>*] **[-n** *<net-addr>*]<br>Displays the AppleTalk route table. | R or M | 1.7 |
| **set\|se AppleTalkRouter\|atr enl\|dis**<br>Enables AppleTalk routing. | R | 1.3.1.2 |
| **set-arpage\|saa** [*<time-value>*]<br>Sets the aging time for entries in the AppleTalk ARP table. | R | 1.6.2 |
| **showcfg\|scf**<br>Displays the current AppleTalk Configuration. | R or M | 1.3.1.3 |
| **stats\|s arp\|ddp\|echo [-t]**<br>Displays statistics for AARP, DDP, or ECHO packets. | R or M | 1.10 |
| **stats-clear\|sc arp\|ddp\|echo**<br>Clears statistics for AARP, DDP, or ECHO packets. | R | 1.11 |
| **zone-table\|zt**<br>[*<start-net-addr>*-*<end-net-addr>*]\|[*<zone-name>*]<br>Displays information about active zones. | R or M | 1.4.1.2 |

*R= Root, M=Monitor.

# 1.3 GETTING STARTED

To set up the PowerHub system for AppleTalk routing, perform these steps:

(1) Enable the AppleTalk subsystem:

- Allocate memory for AppleTalk routing. (See Section 1.3.1.1.)
- Enable AppleTalk routing. (See Section 1.3.1.2.)

(2) Assign AppleTalk zone names to PowerHub segments. (See Section 1.4.1.)

(3) Assign AppleTalk network (interface) addresses to PowerHub segments. (See Section 1.5.1.)

(4) Save your AppleTalk configuration. (See Section 1.3.1.4.)

## 1.3.1 Enabling the AppleTalk Subsystem

Before you can use the AppleTalk subsystem, you must allocate sufficient main memory (DRAM) for the PowerHub system to run the AppleTalk routing subsystem and enable AppleTalk routing.

### 1.3.1.1 Allocating Memory

Before you can use the AppleTalk subsystem, you must allocate a portion of the PowerHub main memory (DRAM) for use by the AppleTalk subsystem. Regardless of how much main memory your PowerHub system contains, you must allocate memory specifically for use by the AppleTalk subsystem.

> **NOTE**: FORE Systems recommends that you allocate memory for the AppleTalk subsystem immediately after you boot the PowerHub system to ensure that the memory you request is available. For more information, see the *PowerHub Installation and Configuration Manual* for your PowerHub system.
>
> You cannot unallocate memory. To free allocated memory, make sure the configuration file does not contain a **`main getmem`** command, then reboot the software.

To allocate memory for the AppleTalk subsystem, issue the following command:

```
main getmem atalk
```

### *1.3.1.2 Enabling AppleTalk Routing*

After you allocate memory, you need to enable AppleTalk routing. Use the **set AppleTalkRouter** command to enable AppleTalk routing:

**set|se AppleTalkRouter|atr enl|dis**

where:

**AppleTalkRouter|atr** Indicates that you are enabling (or disabling) AppleTalk routing on the PowerHub system.

**enl|dis** Specifies whether you are enabling or disabling AppleTalk routing. The default is **dis**.

Here is an example of this command:

```
4:PowerHub:atalk# set atr enl
AppleTalk Routing:  Enabled
```

### *1.3.1.3 Displaying the Current Configuration*

Enter the **showcfg** command to verify that memory is allocated for the **atalk** subsystem and that AppleTalk routing is enabled. This command also displays the aging time for AARP entries. (See Section 1.6.2 on page 21.)

Here is an example of the information listed by this command:

```
5:PowerHub:atalk# showcfg
AppleTalk configuration:
AppleTalk Router : available ( memory available )
AppleTalk Routing: Enabled
AARP Aging Timer:  60 minutes
```

In this example, the display produced by the **showcfg** command shows the following information:

- Memory has been allocated for the AppleTalk subsystem.
- AppleTalk routing is available and enabled.
- The aging time for learned AARP entries is 60 minutes.

### *1.3.1.4 Saving Your AppleTalk Configuration*

After you verify your AppleTalk configuration, you can save the configuration using the **mgmt savecfg** *<file-name>* command or the **tftp svcfg** *<file-name>* command, where *<file-name>* is the configuration file name. When you save your current configuration, the modifications you make to use the AppleTalk subsystem are available next time you reboot the PowerHub system. For information about this command, see the *Installation and Configuration Manual* for your PowerHub system.

# 1.4 CONFIGURING POWERHUB SEGMENTS FOR APPLETALK

Before the PowerHub system can route AppleTalk packets, you must assign the appropriate zone names and network addresses to one or more PowerHub segments. Use the zone commands and interface commands to configure PowerHub segments for use with your AppleTalk networks.

## 1.4.1 The Zone Commands

The PowerHub software uses ZIP (Zone Information Protocol) to maintain a zone table that contains zone names associated with the PowerHub segments. Use the zone commands to add, display, or delete zone names.

### *1.4.1.1 Adding a Zone Name*

Use the **`add-zone`** command to assign a zone name to a segment. A *zone name* is an alphanumeric string up to 32 characters in length. You can assign a different zone name to each segment, assign multiple zone names to the same segment, or assign the same zone name to multiple segments. Zone names are not required for non-seed segments. Moreover, for non-seed segments, the assigned zone names are not used. Assigned zone names are used for seed segments.

The zone name you assign to a PowerHub segment is used by the segment when it attempts to come up as a seed segment. Unless a conflict occurs over the use of the segment as a seed segment, the zone name becomes active for that segment.

You can use blank spaces in zone names. You can use blank spaces at the beginning, inside, or at the end of a zone name.

To add a zone name that contains a leading or trailing blank(s), use double quotes around the entire zone name, including the blank(s).

Here is the syntax for the **`add-zone`** command:

**`add-zone|az`** *`<seg-list>`***`|all`** *`<zone-name>`*

where:

*`<seg-list>`***`|all`** Specifies the PowerHub segment numbers to which you want to assign an AppleTalk zone name. You can list individual segments, specify a range of segments, or specify **`all`** for all segments.

*`<zone-name>`* Specifies the zone name you want to assign to the specified segment. Zone names are not case sensitive. (For example, the zone names `ADMINISTRATION` and `administration` are regarded by AppleTalk as identical.)

In the example that follows, the **`add-zone`** command is used to assign the AppleTalk zone name `Accounting` to segment numbers 1, 3, 4, and 5.

```
8:PowerHub:atalk# add-zone 1,3-5 Accounting
Okay
```

Here is an example of how to add a zone name that contains a leading blank. In this example, the zone name also contains an internal blank

```
1:PowerHub:atalk# add-zone 1 “ Anh Vu”
Okay
```

When you display AppleTalk zone names on the PowerHub system, the names that contain leading or trailing blanks are displayed with quotation marks to show the locations of the blanks.

Here is an example of how zone names that contain blanks are displayed.

```
2:PowerHub:atalk# config-zone-table
CONFIGURED APPLETALK ZONES
    Port          Zone
    ----          ---------------------
    5            " Anh Vu"
3:PowerHub:atalk# name-table
Object Name    ObjectType     Zone
POWERHUB       Router       “ Anh Vu”
```

When you display the zone name in the Chooser on a Macintosh, the blank spaces appear in the zone name but the quotation marks are not displayed.

### *1.4.1.2 Displaying Zone Information*

You can display information for configured zones or for active zones. A *configured zone* is a zone created using the **`add-zone`** command. (See Section 1.4.1.1.) An *active zone* is a zone name that is actively being used on the AppleTalk internet. An active zone can be either a configured zone or a *learned* zone. A *learned zone* is a zone entry learned by the PowerHub software from other routers.

#### Configured Zones

Use the **`config-zone-table`** command to display a list of configured zone names assigned to PowerHub segments. Here is the syntax for this command.

**`config-zone-table|czt [`***`<seg-list>`***`|all]`**

where:

*`<seg-list>`***`|all`** Optionally specifies the segments for which you want to list the configured zone names. You can list individual segments, specify a range of segments, or specify **`all`** for all segments.

In the example that follows, the **`config-zone-table`** command is used to display zone names for segments 1 and 2.

```
11:PowerHub:atalk# config-zone 1,2
CONFIGURED APPLETALK ZONES
       Port          Zone
       1             Test_zone
       2             Test_zone
```

### Active Zones

Use the **`zone-table`** command to display information about active zones (both configured and learned). The **`zone-table`** command shows the network address and the name for each currently active AppleTalk zone that is known to the PowerHub software. In addition, the table indicates whether the zone that is active on a particular network is that network's default zone. Note that configured zone names that are not in use are not listed.

Here is the syntax for this command:

**`zone-table|zt`**

    **`[`***`<start-net-addr>-<end-net-addr>`***`]|[`***`<zone-name>`***`]`**

where:

*`<start-net-addr>-<end-net-addr>`***`|`***`<zone-name>`*

Optionally specifies a specific range of network addresses or a specific zone name. If you do not specify addresses or a zone name, the **`zone-table`** command displays all entries for the PowerHub system.

In the example that follows, the *`<zone-name>`* argument is used to display only the networks on which the zone name "FORE Systems" is active. The asterisks (`**`) to the left of the first network address range indicate that the zone name listed under `Zone` is the default zone name for that network.

```
13:PowerHub:atalk# zone-table FORE Systems
    Net          Zone
**  2-2          FORE Systems
    3-3          FORE Systems
```

Each network has one and only one default zone name. However, the same zone name can be used in more than one network, and can be the default zone name in more than one network.

### *1.4.1.3 Deleting a Configured Zone*

Use the **`del-zone`** command to delete a configured zone name from one or more segments. Here is the syntax for this command:

**`del-zone|dz`** *`<seg-list>`***`|all`** *`<zone-name>`***`|all`**

where:

*`<seg-list>`***`|all`** Specifies the segments from which you want to delete a configured zone. You can list individual segments, specify a range of segments, or specify **`all`** for all segments.

*`<zone-name>`***`|all`** Specifies the zone name you want to delete. You can specify an individual zone name or specify **`all`** for all zone names assigned to the segments you specify.

When you remove a configured zone name, the name disappears from the configured zone table. (Use the **`config-zone-table`** command to display this table.)

> **NOTE**: When you use the **`del-zone`** command to remove a configured zone name, the change is immediately apparent in the Configured-Zone table, but does not affect zone names on interfaces that are currently up. The change can affect an interface if that interface is capable of seeding, and the segment on which the interface is defined is brought down, then back up.

Here are some examples of the use of the **`del-zone`** command. At command prompt 14, a specific zone name (`FORE Systems`) is deleted from specific segments (2 and 4). At prompt 15, a specific zone name (`FORE Systems`) is deleted from *all* segments. At prompt 16, all zone names are deleted from a specific segment (4). At prompt 17, *all* zone names are deleted from *all* segments.

```
14:PowerHub:atalk# del-zone 2,4 FORE Systems
Okay
15:PowerHub:atalk# del-zone all  FORE Systems
Okay
16:PowerHub:atalk# del-zone 4 all
Okay
17:PowerHub:atalk# del-zone all all
Okay
```

## 1.5 CONFIGURING APPLETALK INTERFACES

After you assign zone names to one or more PowerHub segments, you must then assign network addresses to each of these segments. Each network address consists of:

- Network address range.
- Combination of *`<net-addr>.<node-addr>`*.[1]
- Optionally, the default zone name.

### 1.5.1 Adding an Interface (Network Address)

Use the **`add-interface`** command to assign a network address to one or more PowerHub segments.

You can assign a different network address to each segment, or you can assign the same network address to multiple segments. When you assign the same network address to more than one segment, you create a VLAN (virtual LAN), a network that spans two or more physical segments. A *VLAN* lets you increase the effective bandwidth of an

[1] In some books, this combination of net address and node address is called a "port node address," an "AppleTalk protocol address," or a "DDP address," depending upon the context. This manual and other PowerHub documentation uses the term "network address" to refer to this combination.

AppleTalk network without creating additional network numbers. See Appendix D in the *PowerHub Software Manual, V 2.6* for information about the VLAN feature.

Here is the syntax for the **`add-interface`** command:

```
add-interface|ai
    <seg-list> <start-net-addr>-<end-net-addr>|-n
    <net-addr>.<node-addr> [<default-zone-name>]
```

where:

*`<seg-list>`*

Specifies the segment numbers to which you want to assign an AppleTalk network address. You can list individual segments, specify a range of segments, or specify **`all`** for all segments.

*`<start-net-addr>-<end-net-addr>`*

Specifies the beginning and ending network addresses you want to assign to the segment(s). When devices attached to the segment become active, they are dynamically assigned one of the network address numbers within the range you specify.

You can specify a range from **`1`** through **`65023`**.

> **NOTE**: To configure a segment as a non-seed segment, specify a network address range of 0-0. Do not specify a network address following the address range.
>
> If you want to create multiple non-seeding segments, you must issue a separate **`add-interface`** command for each net. If you specify multiple segments with the same command, you create a VLAN.
>
> To configure a segment for a non-AppleTalk (backbone) net, specify **`-n`**, rather than an address range. Do not specify a network address. A backbone net connects routers; nodes are not directly attached to the net.

*`<net-addr>.<node-addr>`*

Specifies the network address you are assigning to the specified segment. The value you specify for *`<net-addr>`* must be within the range specified by *`<start-net-addr>-<end-net-addr>`*.

For *`<node-addr>`*, you can specify a range from **`1`** through **`253`**.

> **NOTE**: Do not use this argument if you are configuring a segment as a non-seed segment or for a non-AppleTalk (backbone) net.
>
> Node addresses 254 and 255 are reserved AppleTalk node addresses for EtherTalk; do not use these addresses. If you attempt to use these addresses, an error message is displayed.

*<default-zone-name>* Optionally specifies the default zone name. This name must already be present in the Configured-Zone table. To display the Configured-Zone table, issue the **configured-zone-table** command.

Here are some examples of the use of the **add-interface** command. In the first example, the network address range 220 through 500 is assigned to segment 1. The network address "220.150" indicates the specific AppleTalk node to which segment 1 is assigned:

```
18:PowerHub:atalk# add-interface 1 220-500 220.150
Port 1 Range 220-500 DDP Addr 220.150 Added
Configured as potential seed for this net.
```

The following example shows the command used to configure segment 5 as a non-seed segment. (Note that no network address range or network address is specified.)

```
19:PowerHub:atalk# add-interface 5 0-0
Port 5 Range 0-0 Added
Configured as non-seeding port.
```

### 1.5.2 Displaying Network Address Information

You can display information about PowerHub segments assigned to an AppleTalk network address using the **interface-table** command.

Here is the syntax for this command:

**interface-table|it**
**[-p** *<seg-list>***|all]**
**|[-n** *<start-net-addr>*-*<end-net-addr>***]**
**|[-z** *<zone-name>***]**
**|[-a]**

where:

**-p** *<seg-list>***|all** Specifies the segment numbers for which you want to display the AppleTalk network addresses. You can list individual segments, specify a range of segments, or specify **all** for all the segments that have AppleTalk interfaces.

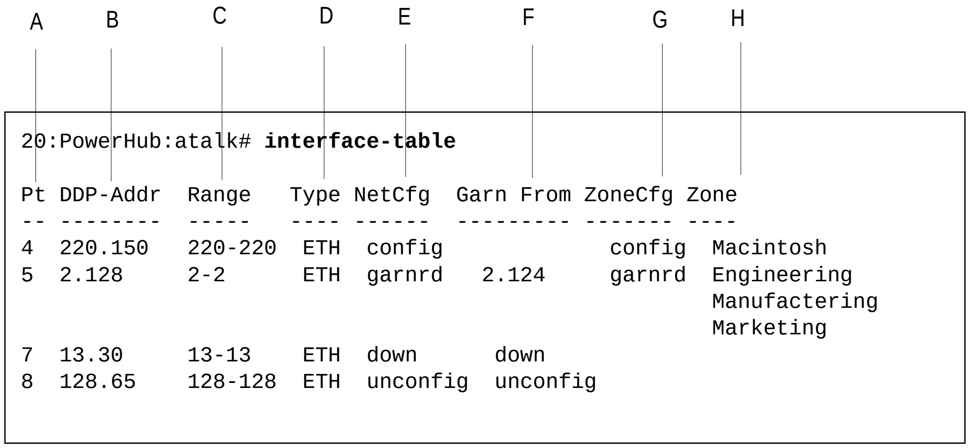
**-n** *<start-net-addr>*-*<end-net-addr>*

Specifies the beginning and ending network addresses for which you want to display information.

**-z** *<zone-name>* Specifies the zone name for which you want to display network address information.

**-a** Lists all configured and non-configured segments.

Here are some examples of the use of the **interface-table** command. In the first example, no arguments are used with the command. Network address information is shown for all segments that have AppleTalk interfaces. Only two AppleTalk network addresses are assigned to PowerHub segments. Note that more than one zone can be associated with a segment. In this example, three zone names are listed for segment 5.



```
20:PowerHub:atalk# interface-table

Pt DDP-Addr  Range   Type NetCfg  Garn From ZoneCfg Zone
-- --------  -----   ---- ------  --------- ------- ----
4  220.150   220-220  ETH  config             config  Macintosh
5  2.128     2-2      ETH  garnrd   2.124     garnrd  Engineering
                                                      Manufactering
                                                      Marketing
7  13.30     13-13    ETH  down      down
8  128.65    128-128  ETH  unconfig  unconfig
```

The table displayed by the **interface-table** command shows the following information:

A The `Pt` column lists the segment numbers.

B The `DDP-Addr` column lists the net address for each segment to which a net address has been assigned. In this example, segments 4 and 5 are assigned AppleTalk net addresses.

C The `Range` column lists the net address range assigned to each AppleTalk segment.

D The `Type` column indicates the media type (in this case, “ETH,” or Ethernet).

E The `NetCfg` column indicates whether the segment was a seed segment (making the hub a seed router) for the network assigned to the segment, or learned the network information from another router in the net.

The `NetCfg` column indicates one of four states: `config`, `unconfig`, `garnrd`, or `down`. The initial state is `unconfig`. If a segment is the seed segment for a network, `config` soon appears under the `NetCfg` column. If the segment is not a seed segment, it instead relies upon another router for seed information. In such a case, when the segment has learned the network address from another router, the state of the segment changes from `unconfig` to `garnrd`. If the segment is not configured as a seed segment and there is no other router on the network, the state remains `unconfig`.

If the state remains `unconfig`, the PowerHub software is unable to find a seed for the segment. Check the connections joining the segment to the seed router. If the connections are working properly, the problem might be in the seed router itself.

If a segment has been configured but is attached to a router that is not turned on, or if a segment is attached to a working router but the segment has been either disabled or has not been added to a zone, the segment is listed as `down`.

If the state is `-cfg`, the segment is part of an AppleTalk VLAN and has gone down. The other segments in the VLAN might still be up.

F The `Garn From` column indicates the seed router from which the PowerHub system got its configuration. If the PowerHub system is the seed router, the `Garn From` field is blank.

G The `ZoneCfg` column indicates whether the interface is a seed router for the zone associated with the segment. Possible states are `config`, `unconfig`, `garnrd`, or `down`. See the descriptions for `NetCfg`.

H The `Zone` column lists the active zone(s) for the segment.

In the following example, the **`-z`** argument is used to limit the display to entries for the specified zone name (in this case, `FORE Systems`):

```
21:PowerHub:atalk# interface-table -z FORE Systems

Pt DDP-Addr  Range   Type NetCfg  Garn From ZoneCfg Zone
-- --------  -----   ---- ------  --------- ------- ----
5  2.128      2-2     ETH  garnrd   2.125     garnrd  FORE Systems
```

**NOTE**: If the interface table displays zeros under the `DDP-Addr` and `Range` columns, or "`down`" for the `NetCfg` and `ZoneCfg` columns, the segment may be down. If the segment is up, check if AppleTalk routing is enabled. See Section 1.3.1.2 for information on enabling AppleTalk routing.

### 1.5.3 Displaying All Interfaces

To display information about all the AppleTalk interfaces configured on the PowerHub system, issue the following command:

**`configured-interface-table|cit`**

When you issue this command, a table such as the following is displayed:

```
1:PowerHub:atalk# cit
Port        Net range            DDP Address
----        ---------            -----------

Individual Networks:
-------------------

 1          100-100              100.101
 2          200-200              200.201

Virtual LAN Configuration:
-------------------------
 3          300-600              300.103
 4          300-600              300.103
 5          300-600              300.103
 6          300-600              300.103
------------------------------------------------
```

The `Port`, `Net range`, and `DDP Address` headers at the top of the display label the types of items displayed in the table columns. In the `Individual Networks` section, AppleTalk interfaces assigned to single segments are listed. In this example, two AppleTalk interfaces have been defined on single segments:

- On segment 1, an interface with the network range 100-100 and the DDP address 100.101 has been defined.
- On segment 2, an interface with the network range 200-200 and the DDP address 200.201 has been defined.

In the `Virtual LAN Configuration` section, the interfaces defined as parts of a VLAN are listed. In this example, four interfaces are listed, all of which have the same network range and DDP address; all four interfaces belong to the same VLAN.

### 1.5.4 Deleting a Network Address

Use the **`del-interface`** command to remove an AppleTalk network address from a PowerHub segment:

**`del-interface|di [-a]`** *`<seg-list>`***`|all`**

where:

**`-a`** Deletes the AppleTalk network address from a segment(s).

**NOTE**: Unless you use the **`-a`** argument, you must specify each segment to which a network is assigned in order to delete a network assigned to multiple segments.

*`<seg-list>`***`|all`** Specifies the segments from which you want to delete the assigned AppleTalk network address. You can list individual segments, specify a range of segments, or specify **`all`** for all segments.

**NOTE**: If you delete an AppleTalk network address, or change or delete the zone name with which the deleted address was associated, we recommend that you wait a minimum of 15 minutes following the zone name change before re-adding the address. This time is needed by the devices in the AppleTalk internet to exchange update information about the network address and zone name changes.

Here is an example of the use of the **`del-interface`** command. In this example, the interface table is displayed to show which interfaces are defined, then the unwanted interfaces are deleted.

```
22:PowerHub:atalk# interface-table

Pt DDP-Addr  Range    Type NetCfg ZoneCfg Zone
1
2
3
4  220.150   220-230  ETH config  config  Macintosh
5  2.128     2-2      ETH garnrd  garnrd  FORE Systems
6  220.150   220-230  ETH config  config  Macintosh
7  220.150   220-230  ETH config  config  Macintosh
8  220.150   220-230  ETH config  config  Macintosh
9
10
11
12

23:PowerHub:atalk#  del-interface 4,6-8
Okay
```

In the example, the network address associated with segments 4, 6, 7 and 8 is deleted. Because the optional **`-a`** argument is not used, all the segments with which the network address is associated must be specified.

The following example uses the **del-interface** command with the **-a** argument to delete the same network address:

```
24:PowerHub:atalk# del-interface -a 6
Okay
```

When you use the **-a** argument, the network address is deleted from all segments to which it is assigned. In this example, network address 220.150, associated with segment 6, is deleted from segment 6 as well as segments 4, 7, and 8. Note that this command performs exactly the same job as the command in the previous example.

## 1.6 USING THE AARP TABLE

The PowerHub system uses AARP (AppleTalk Address Resolution Protocol) to create and maintain a table of translations between MAC-layer node addresses and AppleTalk node addresses. The AARP table enables PowerHub software to look up the MAC-layer address of another device (node, router, and so on) based on the device's AppleTalk address. Entries in the AARP table facilitate transmission of packets from the PowerHub software (acting as an AppleTalk router) to the devices for which MAC-layer addresses are listed. These entries are either static or learned:

*Static entry* An entry that is created when you assign an AppleTalk network address to a PowerHub segment. Each time you assign a network address to a segment using the **add-interface** command, the PowerHub system automatically makes a corresponding entry in the AARP table. These entries cannot be deleted unless the corresponding network address is deleted.

*Learned entry* An entry that the software automatically adds to the AARP table when it learns about a node address from another managed PowerHub or other AppleTalk router, or learns of the node address directly from one of its own segments. The PowerHub software deletes learned entries when they are inactive for the *AARP aging time*.

For information on the AARP aging time, see Section 1.6.2. For each entry, the hub's AARP table lists the:

- DDP address of the node (also known as AppleTalk node address).
- Type of connection the segment has. There are four types of connections:

  `Local` Indicates a device is directly attached to the segment.

  `Router` Indicates the route was dynamically learned. Also indicates another AppleTalk router.

`Bcast` Indicates the entry in the AARP table is broadcast to all devices in the network. A broadcast packet is denoted by a node address of **`255`**.

*blank* Indicates a learned address, one that is added by the software. Blank entries also indicate that a node, not a router, is attached.

- MAC-layer address.
- Segment to which the node is attached.

### 1.6.1 Displaying AARP Entries

Use the **`arp-table`** command to display the entries in the AARP table. Here is the syntax for this command:

**`arp-table|at [`***`<net-addr>`***`.`***`<node-addr>`***`]`**

where:

*`<net-addr>`***`.`***`<node-addr>`* Specifies the network address for which you want to display AARP entries.

Here are some examples of the use of the **`arp-table`** command. In the first example, the command is entered without an argument. The table displayed lists all AARP entries, both static entries and learned entries, for this PowerHub system.

```
25:PowerHub:atalk# arp-table
  ARP TABLE:
DDP Address  Type    Mac Address         Port
-----------  ------  -----------------   -------------
2.5          Local   00-00-ef-02-41-50   2
2.22                 00-00-94-20-5f-82   2
2.255        BCast   09-00-07-ff-ff-ff   2
111.1        Local   00-00-ef-02-41-50   3,4
111.22               00-00-94-21-fd-1c   3
111.56               00-00-94-21-f2-43   4
111.255      BCast   09-00-07-ff-ff-ff   3,4
5.1          Local   00-00-ef-02-41-50   5
5.255        BCast   09-00-07-ff-ff-ff   5
```

In the following example, a specific value for *`<net-addr>`***`.`***`<node-addr>`* is given:

```
26:PowerHub:atalk# arp-table 2.5

DDP Address  Type    Mac Address         Port
-----------  ------  -----------------   -------------
2.5          Local   00-00-ef-02-41-50   2
```

You can specify a wildcard (**`*`**) in place of *`<net-addr>`* or *`<node-addr>.`* In the following example, all DDP addresses with the *`<net-addr>`* "2" are displayed.

```
 27:PowerHub:atalk# arp-table 2.*
   ARP TABLE:
DDP Address  Type    Mac Address         Port
-----------  ------  -----------------   -------------
2.5          Local   00-00-ef-02-41-50   2
2.22                 00-00-94-20-5f-82   2
2.255        BCast   09-00-07-ff-ff-ff   2
```

> **NOTE**: If the AARP table is blank, AppleTalk routing might not be enabled. Use the **`showcfg`** command to verify that routing is enabled. (See Section 1.3.1.3 on page 8.) If routing is not enabled, see Section 1.3.1.2.

### 1.6.2 Setting the AARP Aging Time

You can configure the PowerHub software to maintain the AARP table by specifying the amount of time learned entries can remain inactive in the AARP table before being removed by the software. This time limit is the *AARP aging interval* and is independent of the aging time for routing table entries.

Use the **`set-arpage`** command to set the number of minutes a learned AARP entry can be inactive before the PowerHub software deletes it from the AARP table.

Here is the syntax for this command:

**`set-arpage|saa [`***`<time-value>`***`]`**

where:

| | |
|---|---|
| *`<time-value>`* | Specifies the number of minutes that inactive entries remain in the AARP table. The minimum aging time is **`3`** minutes. The default is **`60`** minutes. |

At command prompt 28 in the example that follows, the **`set-arpage`** command is used with the *`<time-value>`* argument to change the AARP aging time to 30 minutes. At command prompt 29, the **`set-arpage`** command is used without this argument to display the current AARP aging time.

```
28:PowerHub:atalk# set-arpage 30
ARP Age changed to 30 minutes.
29:PowerHub:atalk# set-arpage
ARP Age set to 60 minutes.
```

### 1.6.3 Clearing the AARP Table

Use the **`arp-tableclear`** command to clear all learned entries from the AARP table. Here is an example of the use of this command:

```
30:PowerHub:atalk# arp-tableclear
Okay
```

## 1.7 DISPLAYING ROUTE INFORMATION

Each PowerHub system serving as a router in an AppleTalk internet uses RTMP (Routing Table Maintenance Protocol) to maintain a table of information about other AppleTalk routes throughout the internet.

Use the **`route-table`** command to display the AppleTalk route table. For each route, the route table lists the:

- Destination network address.
- Network address of the next hop (if the route is to another router).
- Segment number associated with the next hop.
- Cost (number of hops, or intermediate routers).
- State (`good`, `suspect`, or `bad`).

Periodically, each AppleTalk router (including other PowerHub systems serving as AppleTalk routers) broadcasts RTMP packets through each of its segments configured for AppleTalk to the other AppleTalk routers and nodes adjacent to it. As a result, each router in an AppleTalk network always has a current list of routes to the other networks.

Here is the syntax for this command:

**`route-table|rt [-c|-r] [-t] [-p`** *`<seg-list>`*  **`] [-n`** *`<net-addr>`***`]`**

where:

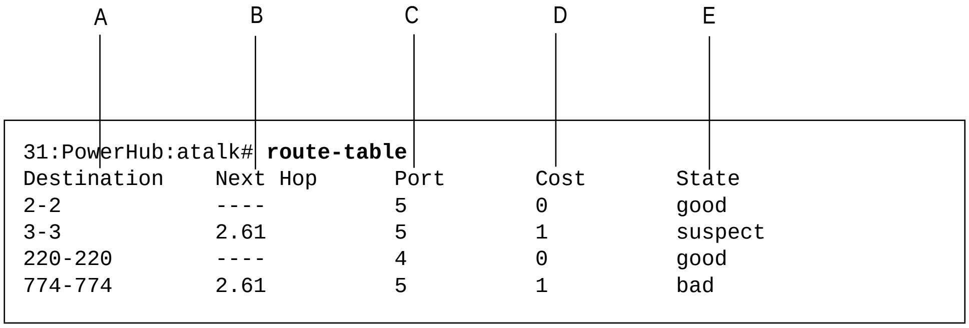
**`-c|-r`** Restricts the display to only directly connected entries (**`-c`**) or RTMP entries (**`-r`**).

**`-t`** Displays the total number of entries in the route table.

**`-p`** *`<seg-list>`* Specifies the segments for which you want to display route information.

**`-n`** *`<net-addr>`* Specifies the AppleTalk net address for which you want to display route information.

Here is an example of the use of the **`route-table`** command.



```
31:PowerHub:atalk# route-table
Destination     Next Hop      Port       Cost       State
2-2             ----          5          0          good
3-3             2.61          5          1          suspect
220-220         ----          4          0          good
774-774         2.61          5          1          bad
```

In this example, the routes for four destinations are shown:

A Under the `Destination` column is listed the network address range for each route in the routing table.

B The `Next Hop` column labels the network address of the router at the next hop. When a destination is local to the router, the next hop field contains dashes (`----`).

C The `Port` column indicates the segment number through which the route (listed as the destination) can be reached.

D The number under `Cost` indicates how many hops (routers) a packet must pass through to reach the destination.

E The `State` column lists the state of the route.

A route can have one of three states: `good`, `suspect`, or `bad`. Approximately every 10 seconds, the PowerHub software broadcasts an RTMP packet to each adjacent hub to inform these hubs of active (`good`) routes. When the software does not receive this RTMP packet within 20 seconds, it changes the status for the routes from `good` to `suspect`.

After a route becomes `suspect`, the PowerHub software waits an additional 20 seconds to receive the status packet. When the packet is received within 20 seconds, the status is changed from `suspect` back to `good`. If the packet is not received, the status changes form `suspect` to `bad`. When a route's status changes to `bad`, the PowerHub software waits another 20 seconds for an RTMP packet. If the packet still is not received, the bad route is removed from the routing table.

Here is an example of the display produced if you use the **`-c`** argument, which displays entries only for directly connected networks:

```
32:PowerHub:atalk# route-table -c

Destination     Next Hop      Port                  Cost      State
2-2             ----          5                     0         good
220-220         ----          4                     0         good
```

Because the routes listed in this display are for directly connected destinations, no value appears under the `Next Hop` column for either route.

Here is an example of the display produced if you use the **`-c`** and **`-p`** arguments:

```
33:PowerHub:atalk# route-table -c -p 4

Destination    Next Hop      Port        Cost      State
220-220        ----          4           0         good
```

The arguments used to produce this display restrict the information to only those routes that are directly connected and are attached to segment number 4.

> **NOTE**: If the route table is blank, AppleTalk routing might not be enabled. Use the **`showcfg`** command to verify that routing is enabled. (See Section 1.3.1.3 on page 8.) If routing is not enabled, see Section 1.3.1.2.

## 1.8 USING THE ROUTE CACHE

The AppleTalk *route cache* shows, for each segment, the most recently used destination networks. At any time, you can get an at-a-glance picture of AppleTalk routing activity in your network by displaying the AppleTalk route cache.

### 1.8.1 Displaying the Route Cache

Use the **`display-routecache`** command to display the AppleTalk route cache. AppleTalk is the syntax for this command.

**`display-routecache|dc [`***`<seg-list>`***`]`**

where:

| | |
|---|---|
| *`<seg-list>`* | Specifies the segments for which you want to display information in the route cache. If you do not specify a segment, information for all segments is shown. |

Here is an example of the display produced by this command:

```
33:PowerHub:atalk# display-routecache
Appletalk router cache:
Port 01: empty
Port 02: 111.22,111.56
Port 03: 2.22
Port 04: 2.22
Port 05: empty
Port 06: empty
Port 07: empty
Port 08: empty
Port 09: empty
Port 10: empty
Port 11: empty
Port 12: empty
Port 13: empty
Port 14: empty
```

**NOTE**: The contents of the route cache can change quite rapidly. As a result, successive **`display-routecache`** commands can give different results.

### *1.8.2 Flushing the Route Cache*

The **`flush-routecache`** command removes all entries for all segments from the route cache. After the cache is flushed, new entries are again added, using the cache’s “most-recently-used” algorithm. Thus, you can use the **`flush-routecache`** command to ensure that all entries displayed by a subsequent **`display-routecache`** command are fresh.

## 1.9 DISPLAYING NBP INFORMATION

The PowerHub software uses NBP (Name Binding Protocol) to associate names with AppleTalk network numbers, node addresses, socket numbers, and other services. With NBP, you can bind a meaningful name to any service in an AppleTalk internet. For example, you might use NBP to bind the name **“**`Printer1`**”** to a socket number to which a printer is attached.

> **NOTE**: The NBP table maintained by the PowerHub software lists only the objects registered with the PowerHub system.

For each service registered with the PowerHub system, the NBP table lists the:

- Object name.
- Object type.
- Zone in which the object resides.

To display the NBP table, use the **`name-table`** command. Here is an example of the information displayed by this command:

```
34:PowerHub:atalk# name-table
Object Name           ObjectType     Zone
PORT_220.150          Router         Macintosh
POWERHUB              Router         FORE Systems
```

A network administrator used AppleTalk NBP to name the two objects (services) “`PORT_220.150`” and “`POWERHUB`.” Both objects are registered to this PowerHub system as type “`Router`.” They belong to different zones, “`Macintosh`” and “`FORE Systems`,” respectively.

## 1.10 DISPLAYING STATISTICS

During operation of your AppleTalk networks, the PowerHub software collects statistics for AARP (AppleTalk Address Resolution Protocol), DDP (Datagram Delivery Protocol), and AEP (AppleTalk Echo Protocol) packets.

Use the **`stats`** command to display statistics for AppleTalk ARP, DDP, or AEP packets. Here is the syntax for this command:

**`stats|s arp|ddp|echo [-t]`**

where:

**`arp|ddp|echo`** Specifies the type of AppleTalk protocol for which you want to display statistics.

**`-t`** Displays statistics collected since the most recent system reset, rather than those collected since the most recent clear (using the **`stats-clear`** command).

The types of statistics the PowerHub software collects and displays depends upon the protocol type.

Here is an example of information displayed for the AARP protocol:

```
35:PowerHub:atalk# stats arp
ARP Statistics:

Requests received:            992
Replies received:             296
Invalid packets received:     0
Requests sent:                79
Replies sent:                 0
```

Here is an example of the information displayed for the DDP protocol:

```
36:PowerHub:atalk# stats ddp
Out Requests:                 93734
Out Shorts:                   0
Out Longs:                    93734
In Receives:                  82180
Forward Requests:             63849
In Local Datagrams:           78658
No Proto Handler:             0
Out No Routes:                0
Too Short Errors:             0
Too Long Errors:              0
Broadcast Errors:             0
Short DDP Errors:             0
Hop Count Errors:             0
Checksum Errors:              0
Config Address Errors:        0
Config Zone Errors:           0
```

Here is an example of the information displayed for the AEP (echo) protocol:

```
37:PowerHub:atalk# stats echo
Echo requests received:       39596
Echo replies received:        0
Echo requests sent:           0
```

**NOTE**: If a table displayed by the **stats** command contains all zeroes for the statistics amounts, AppleTalk routing might not be enabled. Use the **showcfg** command to verify that routing is enabled. (See Section 1.3.1.3 on page 8.) If routing is not enabled, see Section 1.3.1.2 on page 8.

## 1.11 CLEARING APPLETALK STATISTICS

To clear the statistics collected since the most recent clear, use the **stats-clear** command:

**stats-clear|sc arp|ddp|echo**

where:

**arp|ddp|echo** Specifies the type of AppleTalk protocol for which you want to clear statistics.

## 1.12 TESTING A NETWORK ADDRESS

You can use the **ping** command to test the accessibility of and round-trip delay to any AppleTalk node. This command sends an AEP (AppleTalk Echo Protocol) packet to the specified node. The AEP packet contains an instruction to the receiving device to forward the packet back to the sending PowerHub system, thus verifying receipt of the packet.

To send an AEP packet, use the following command:

**ping** *<net-addr>***.***<node-addr>* **[***<time-out>* **[***<pktsize>***]]**

where:

*<net-addr>***.***<node-addr>*

Specifies the network node to which you want to send the test packet.

*<timeout>* Optionally specifies the number of seconds the PowerHub system waits to receive a reply packet from the specified node. The default is **15** seconds.

*<pktsize>* If you use the *<timeout>* argument, optionally specifies the size of the echo packet you want to send to the node. The packet size is measured in bytes. You can specify a packet size of 64-586 bytes. The default is **64** bytes.

The following example shows the results of the **ping** command when an AEP packet is successfully received by the sending PowerHub system:

```
39:PowerHub:atalk# ping 220.150
220.150 is alive
```

If the target node to which you send an AEP packet is not found, or if the timeout expires before the return packet is received, an error message is displayed.

In such a case, check the route table for the network on which the specified target node resides. If the network is listed in the table, check the configuration for the target node to ensure it has learned the current network and zone-related information. If the route table and target node are okay, check the physical connections between the PowerHub system and the target node.

# 2 IPX Commands

This chapter describes the commands in the **ipx** subsystem and tells you how to use the commands to configure and manage the PowerHub system as an IPX router. You can use the commands in this subsystem to perform the following tasks:

- Allocate main memory for IPX. (See Section 2.4.)
- Display the IPX configuration. (See Section 2.5.)
- Configure an IPX interface. (See Section 2.6.)
- Enable IPX forwarding and set other IPX parameters. (See Section 2.7.)
- Use the IPX Route Table. (See Section 2.8.)
- Use the IPX Route Cache. (See Section 2.9.)
- Use the IPX Server Table. (See Section 2.10.)
- Configure IPX RIP and SAP. (See Section 2.11.)
- Display and clear statistics. (See Section 2.13.)
- Configure the hub to assist IPX broadcast packets. (See Section 2.14.)

## 2.1 ACCESSING THE IPX SUBSYSTEM

To access the **ipx** subsystem, issue the following command at the runtime command prompt:

```
ipx
```

Most of the commands in this chapter assume that you have changed the focus of the command prompt to “`ipx`.” A few commands, such as **`getmem`**, are not in the **ipx** subsystem. This chapter identifies such commands by listing their subsystem name with the command (ex: **`main getmem`**.)

## 2.2 IPX SUBSYSTEM COMMANDS

Table 2–1 lists and describes the **ipx** subsystem commands and their syntax. For each command, the management capability (root or monitor) is listed, as well as the section that contains additional information about the command.

**TABLE 2–1** IPX subsystem commands.

| Command and Description | Capability* | See… |
|---|---|---|
| **add-interface\|ai**<br>*<seg-list> <network>*<br>**[***<mtu>***] [enet\|802.3\|802.2\|snap] [cost** *<cost>***]**<br>Adds an IPX interface. | R | 2.6.2 |
| **add-route\|ar**<br>*<network> <gw-net> <gw-addr> <seg> <hops>*<br>*<ticks>*<br>Adds an IPX route. | R | 2.8.2 |
| **add-server\|as**<br>*<s-type> <s-net> <s-addr> <s-sock> <s-hops>*<br>*<s-name>*<br>Adds an IPX server. | R | 2.10.2 |
| **del-interface\|di** *<seg-list>***\|all** *<network>***\|all**<br>Deletes an IPX interface. | R | 2.6.5 |
| **del-route\|dr** *<network> <gw-net> <gw-addr>*<br>Deletes an IPX route. | R | 2.8.3 |
| **del-server\|ds** *<s-type> <s-name>*<br>Deletes an IPX server. | R | 2.10.3 |
| **display-routecache\|dc [***<seg-list>***]**<br>Displays the IPX route cache. | R or M | 2.9 |
| **flush-routecache\|fc**<br>Flushes the route cache. | R | 2.9.2 |
| **helper add**<br>*<network> <node-addr> <socket> <seg-list>*<br>Adds a helper address to the specified segment(s). | R | 2.14.2 |
| *R= Root, M=Monitor. | | |

**TABLE 2–1** (Continued) IPX subsystem commands.

| Command and Description | Capability* | See… |
|---|---|---|
| **helper delete** *<seg-list>*<br>Deletes a helper address from a specified segment(s). | R | 2.14.4 |
| **helper show**<br>Shows the helper address for all segments. | R or M | 2.14.4 |
| **interface-table\|it [-p** *<seg-list>***] [-n** *<network>***]**<br>Displays the IPX interface table. | R or M | 2.6.4 |
| **rip-control-tbl\|rco add\|a**<br>*<network> <parm-list>***\|all yes\|y no\|n**<br>Controls the mode for sending and receiving RIP and SAP updates. | R | 2.11.3.2 |
| **rip-control-tbl\|rco del\|d** *<network>*<br>Deletes the specified network's **rip-control-tbl** entries. | R | 2.11.4.2 |
| **rip-control-tbl\|rco shows\|s [***<network>***]**<br>Shows the RIP control table entry for the requested network. | R or M | 2.11.2.2 |
| **rip-pset\|rpse**<br>*<seg-list>***\|all** *<parm-list>***\|all yes\|y\|no\|n**<br>Enables IPX RIP on a per-segment basis. | R | 2.11.3.1 |
| **rip-sap-ctrl-type\|rsct [normal\|n vlan\|v]**<br>Controls the mode for sending and receiving RIP and SAP updates. | R | 2.11.1 |
| **rip-showcfg\|rscf [***<seg-list>***]**<br>Displays the current IPX RIP configuration. | R or M | 2.11.2.1 |
| **rip-stats\|rst [-t]**<br>Displays IPX RIP statistics. | R or M | 2.13.2 |
| **rip-stats-clear\|rstc**<br>Clears the IPX RIP statistics. | R | 2.13.2.1 |
| **route-table\|rt**<br>**[-c\|-r] [-p** *<seg-list>* **\| -n** *<network>***] [-t]**<br>Displays the IPX route table. | R or M | 2.8.1 |
| *R= Root, M=Monitor. | | |

**TABLE 2–1** (Continued) IPX subsystem commands.

| Command and Description | Capability* | See… |
|---|---|---|
| **sap-control-tbl\|sco add\|a**<br>*<network> <parm-list>*\|**all yes\|y\|no\|n**<br>Sets SAP parameters for the specified network address. | R | 2.11.3.2 |
| **sap-control-tbl\|sco del\|d** *<network>*<br>Deletes the specified network’s **sap-control-tbl** entries. | R | 2.11.4.2 |
| **sap-control-tbl\|sco show\|s [***<network>***]**<br>Shows the SAP control table for the requested network. | R or M | 2.11.2.2 |
| **sap-pset\|spse**<br>*<seg-list>***\|all** *<parm-list>***\|all y\|yes\|n\|no**<br>Defines IPX SAP options on a per-segment basis. | R | 2.11.3.1 |
| **sap-showcfg\|sscf [***<seg-list>***]**<br>Displays the current IPX SAP configuration. | R or M | 2.11.2.1 |
| **sap-stats\|sst [-t]**<br>Displays the IPX SAP statistics. | R or M | 2.13.3 |
| **sap-stats-clear\|sstc**<br>Clears the IPX SAP statistics. | R | 2.13.3.1 |
| **server-table\|st**<br>**[-s** *<s-type>***] [-p** *<seg-list>***]**<br>**[-n** *<network>***] [-i** *<name>***]**<br>**[-f \| -t \| -a]**<br>Displays the IPX server table. | R or M | 2.10.1 |
| **set\|se** | R | |
| **helper enl\|dis** | | 2.14.1 |
| **ipxForwarding\|xfw enl\|dis** | | 2.7 |
| **large-rip-sap-pkt\|lpkt enl\|dis** | | 2.11.5 |
| **rip-timers\|rtmr**<br>*<rip-bcast-intvl>* **[***<rip-age>***]** | | 2.12.3.1 |
| **sap-timers\|stmr**<br>*<sap-bcast-intvl>* **[***<sap-age>***]** | | 2.12.3.2 |

*R= Root, M=Monitor.

**TABLE 2–1** (Continued) IPX subsystem commands.

| Command and Description | Capability* | See… |
|---|---|---|
| **`type20-forwarding|t20fw enl|dis`**<br>Enables or disables IPX forwarding and the generation of large RIP and SAP packets. Also, use the **`set`** command to specify the SAP and RIP broadcast interval and age timer values, to enable type-20 packet forwarding, and to define IPX helper addresses. | | 2.12.2.1 |
| **`showcfg|scf`**<br>Displays the current settings of IPX parameters. | R or M | 2.5 |
| **`stats|s [ipx|type20] [-t]`**<br>Displays IPX statistics. | R or M | 2.13 |
| **`stats-clear|sc [ipx|type20]`**<br>Clears IPX statistics. | R | 2.13.1.1 |
| **`type20-port-forwarding|tpfw enl|dis`** *`<seg-list>`***`|all`**<br>Enables or disables forwarding of type-20 packets to or from the specified segments. | R | 2.12.2.2 |
| **`type20-port-forwarding|tpfw show`** *`<seg-list>`***`|all`**<br>Displays whether type-20 packets are being forwarded to or from the specified segments. | R or M | 2.12.1 |
| *R= Root, M=Monitor. | | |

## 2.3 GETTING STARTED

To set up the PowerHub system for IPX routing, you must perform the following steps:

(1) Allocate memory for IPX routing. (See Section 2.4.)

(2) Configure IPX interfaces on PowerHub segments. (See Section 2.6.2.)

(3) Enable IPX forwarding. (See Section 2.7.)

(4) (Optional) Configure static IPX routes in the IPX route table. (See Section 2.8.)

(5) (Optional) Configure static servers in the IPX server table. (See Section 2.10.)

(6) Enable forwarding and sending for the IPX RIP (Routing Information Protocol) and SAP (Service Advertising Protocol) as needed. (See Section 2.11.3.)

(7) (Optional) Enable forwarding and sending of large IPX RIP and SAP packets. (See Section 2.11.5.)

(8) (Optional) Enable sending and forwarding of type-20 packets. (See Section 2.12.)

(9) (Optional) Enable automatic segment-state detection, if you want the route state for a segment to reflect the corresponding segment's physical connection state (up or down). For information on enabling automatic segment-state detection, see the *Installation and Configuration Manual* for your PowerHub system.

After setting up IPX routing, you can check connectivity to hosts and other routers using commands on NetWare workstations and servers. For example, the "**`SLIST`**" command on workstations displays all reachable file servers in the IPX internet, and the "**`DISPLAY NETWORKS`**" and "**`DISPLAY SERVERS`**" commands on file servers show all reachable networks and servers. You also can check connectivity from the PowerHub system using the **`server-table`** and **`route-table`** commands in the **ipx** subsystem.

> **NOTE**: The network and server information displayed by workstations, servers, and the PowerHub system might take a few minutes to reflect changes in network configuration because of the relatively slow convergence time of RIP and SAP.

When you are satisfied with your IPX configuration, we recommend that you save the configuration using the **`mgmt savecfg`** *`<file-name>`* command, where *`<file-name>`* is the name of a configuration file. For information on this command and on configuration files, see the *Installation and Configuration Manual* for your PowerHub system.

## 2.4 ALLOCATING MEMORY

Before you can use the **ipx** subsystem, you must allocate a portion of the PowerHub main memory for use by the **ipx** subsystem. Regardless of how much main memory your PowerHub system contains, you must allocate memory specifically for use by the **ipx** subsystem.

> **NOTE**: FORE Systems recommends that you allocate memory for the IPX subsystem immediately after you boot the PowerHub system to ensure that the memory you request is available. For more information, see the *PowerHub Installation and Configuration Manual* for your PowerHub system.
>
> You cannot deallocate memory. To free allocated memory, make sure the configuration file does not contain a **`main getmem`** command, then reboot the software.

To allocate memory for the IPX subsystem, issue the following command:

```
main getmem ipx
```

## 2.5 DISPLAYING THE IPX CONFIGURATION

You can display the current IPX settings on the PowerHub system by issuing the **`showcfg`** command. Here is an example of the information displayed by this command

```
6:GE:ipx# showcfg
IPX Configuration:

IPX Router:                    Memory Available
IPX Forwarding:                enabled
IPX Type20 Packet Forwarding: enabled
IPX Helper Feature:            enabled
Large RIP and SAP Packets:     disabled
RIP broadcast timer interval: 60
SAP broadcast timer interval: 60
RIP aging timer interval:      180
SAP aging timer interval:      180
```

You can set any of the IP configuration items listed in this display.

`IPX Router` Indicates whether main memory has been allocated for the IPX subsystem. (See Section 2.4.)

`IPX Forwarding` Indicates whether IPX forwarding is enabled or disabled. (See Section 2.7.) The default setting is disabled.

`IPX Type20 Packet Forwarding`
Indicates that the hub is configured to forward type-20 IPX packets. (See Section 2.12.) The default setting is enabled.

`IPX Helper Feature`
Indicates the setting of the IPX helper feature. When enabled, this feature allows the hub to forward unknown IPX broadcast packets. (See Section 2.14.)

`Large RIP and SAP Packets`
Indicates whether the hub is enabled to forward large (longer than 576 bytes) IPX RIP and SAP packets. (See Section 2.11.5.) The default setting is disabled.

`RIP broadcast timer interval`
Indicates how often the hub sends RIP broadcasts. (See Section 2.12.3.) The default is 60 seconds.

`SAP broadcast timer interval`
Indicates how often the hub sends SAP broadcasts. (See Section 2.12.3.) The default is 60 seconds.

`RIP aging timer interval`

Indicates how many seconds a learned, unused IPX route can remain in the route table before it is removed by the software's aging mechanism (See Section 2.12.3.) The default is 180 seconds, but if you choose a value other than the default, the RIP aging timer interval is always three times the RIP packet aging interval.

`SAP aging timer interval`

Indicates how many seconds a learned, unused IPX server can remain in the server table before it is removed by the software's aging mechanism. (See Section 2.12.3.) The default is 180 seconds, but if you choose a value other than the default, the SAP aging timer intervals always three times the SAP packet aging interval.

You can configure any of the IPX configuration items listed in this display. Sections in this chapter describe the commands you use to set these items.

> **NOTE**: For information on displaying RIP and SAP settings, see Section 2.11.2.

### 2.5.1 Saving Your IPX Configuration

After you verify your IPX configuration, you can save the configuration using the **`mgmt savecfg`** *`<file-name>`* command or the **`tftp svcfg`** *`<file-name>`* command, where *`<file-name>`* is the configuration file name. When you save your current configuration, the modifications you make to use the IPX subsystem are available next time you reboot the PowerHub system. For information about this command, see the *Installation and Configuration Manual, V 2.6* for your PowerHub system.

## 2.6 CONFIGURING IPX INTERFACES

To configure an IPX interface, use the **`add-interface`** command (see Section 2.6.2). This section describes how to use the **`add-interface`** command. Before you begin configuring the IPX interfaces, we recommend you read the considerations in Section 2.6.1.

### 2.6.1 Considerations

- IPX network addresses have two parts, a 32-bit *network number* and a 48-bit *node number*. The node number is generally the same as the node's MAC-layer hardware address. Before enabling IPX routing, you must assign a 32-bit IPX network number to each PowerHub segment that will route IPX packets. The range for valid network numbers is **`1`** through **`0xfffffffe`**
- A connection to a network segment is sometimes called an *interface*. The IPX software maintains an "interface table," discussed in Section 2.6.4, to keep track of network/segment assignments. Use the **`add-interface`**, **`interface-table`**, and **`del-interface`** commands to configure IPX networks on segments.
- We recommend that you assign IPX network numbers in accordance with your own organization's guidelines. If your organization does not have such guidelines, refer to a NetWare reference manual for suggestions.
- The data portion of an IPX packet can be "encapsulated" in one of four different formats before being transmitted on an Ethernet segment. (See *Appendix C in the PowerHub Software Manual, V 2.6* for information on these formats.) The network administrator establishes an encapsulation for each IPX network. The choice of encapsulation is established in end-user stations, in servers, and in routers that connect multiple IPX networks, by a variety of means:

  *End-stations* The encapsulation is usually specified in the station's NET.CFG file.

  *Servers* The encapsulation is determined by commands in the server's AUTOEXEC.NCF file.

  *Routers* In the PowerHub software, the encapsulation is determined by a parameter of the **`add-interface`** command, explained in Section 2.6.2.
- The encapsulation specified for a PowerHub interface ***must*** match the encapsulation used by IPX servers and workstations on the network attached to that interface.
- In NetWare versions prior to 3.1, Ethernet stations use a default encapsulation type of 802.3. As a result, selecting the **`802.3`** encapsulation in the PowerHub configuration generally works in such pure Novell Ethernet networks.

- In a mixed environment, if any other protocols use 802.3 encapsulation (ex: TCP/IP on certain Hewlett-Packard workstations), then 802.3 encapsulation ***must not*** be used for IPX. There is no way to distinguish between IPX and other packets in this case. The **`enet`** encapsulation is often the best choice in such environments.
- VAX hosts running NetWare for VMS use Ethernet II encapsulation, as do many UNIX hosts. In such cases, we recommend that you use the **`enet`** encapsulation.

Some networks use either 802.2 or 802.2/SNAP encapsulation to help support the coexistence of multiple logical links and protocols across heterogeneous physical networks. In this case, we recommend that you use the **`802.2`** or **`snap`** encapsulation. Individual situations might vary.

For proper network operation, the choice of encapsulations within each IPX network must be consistent. However, servers and routers have the ability to translate packets from one encapsulation into another when forwarding packets between networks.

**NOTE**: You also can configure the PowerHub system to perform IPX translation bridging. However, IPX translation bridging is independent of IPX routing—they are mutually exclusive. We recommend that you do not enable both IPX translation bridging and IPX routing. If both IPX translation bridging and IPX routing are enabled, IPX routing takes precedence over IPX translation bridging.

### 2.6.2 VLAN Support

PowerHub software version 2.6 provides full IPX Virtual LAN (VLAN) support. These capabilities are described in Appendix D of the *PowerHub Software Manual, V 2.6*.

### 2.6.3 Adding an IPX Interface

Use the **`add-interface`** command to assign an IPX interface (sometimes referred to as a network number) to one or more PowerHub segments. When you add an interface, the software also makes an entry in the route table, to show that the network is directly connected to the specified segment. (See Section 2.8.)

Here is the syntax for the **`add-interface`** command.

```
add-interface|ai
    <seg-list> <network> [<mtu>]
    [enet|802.3|802.2|snap] [cost <cost>]
```

where:

*`<seg-list>`* Specifies the segment number(s) to which you are assigning the IPX interface. You can specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.

**NOTE**: If you specify more than one segment number per interface, you are creating an IPX interface for a virtual LAN (VLAN). See Appendix D in the *PowerHub Software Manual, V 2.6* for information on IPX VLANs.

*<network>* Specifies an IPX network number. Specify a hexadecimal number in the range from **1** through **ffffffe**.

*<mtu>* Specifies the maximum transmission unit (number of octets) for packets forwarded on this segment. Specify a number in the range from **576** through **1500**. The default is **576**.

> **NOTE**: The *<mtu>* parameter actually is not used in the current version of PowerHub software, but is provided for compatibility with future IPX protocol requirements.

**enet|802.3|802.2|snap**

Specifies the type of encapsulation to be used for IPX packets sent and received on this segment. You can specify one of the following encapsulation types:

| | |
|---|---|
| **enet** | Ethernet II. |
| **802.3** | IEEE 802.3. |
| **802.2** | IEEE 802.3 with 802.2 (LLC) header. |
| **snap** | IEEE 802.3 with 802.2 (LLC) and SNAP headers. |

IPX packets (such as RIP and SAP packets) generated on this segment have the specified encapsulation. IPX packets received on a segment with one encapsulation are translated into other encapsulations as required for forwarding to other segments. The default is **802.3**.

**cost** *<cost>* Specifies an additional cost (extra hops) of using the interface. You can specify a cost in the range 1–14. When the PowerHub software reports this subnet using RIP, it adds the additional cost to the reported metric.

Here are some examples of the use of this command.

```
1:PowerHub:ipx# add-interface 1 1001 enet
Port 1, Network 1001, MTU 576, Cost 0, Frame type Ethernet II
Added
2:PowerHub:ipx# add-interface 6 2002 enet cost 1
Port 6, Network 2002, MTU 576, Cost 1, Frame type Ethernet II
Added
```

The first command creates an IPX interface on segment 1. Because this interface is intended to be used as the primary route to the PowerHub system from a router, no cost is specified.

The second command creates an IPX interface on segment 6. However, a cost has been added to this interface. The PowerHub RIP software adds this cost to the route when it reports it to the other routers attached to segment 6.

Here is an example of the **`add-interface`** command used to add an IPX network to more than one PowerHub segment. This command creates an IPX VLAN.

```
23:PowerHub:ipx# add-interface 1-3 55ccdd55 576 802.2
Port 1, Network 55ccdd55, MTU 576, Frame type 802.2
Added
Port 2, Network 55ccdd55, MTU 576, Frame type 802.2
Added
Port 3, Network 55ccdd55, MTU 576, Frame type 802.2
Added
```

### 2.6.4 Displaying the Interface Table

You can view the network numbers assigned to segments by using the **`interface-table`** command. Here is the syntax for this command.

**`interface-table|it [-p`** *`<seg-list>`***`] [-n`** *`<network>`***`]`**

where:

**`-p`** *`<seg-list>`* Specifies the segments for which you want to display IPX interface information. If you specify a list or range of segments, information is shown for only those segments that have IPX interfaces.

**`-n`** *`<network>`* Specifies the IPX network for which you want to display information.

The display includes the segment state—`UP`, if the segment is up, or `DOWN`, if you have disabled the segment or if the automatic segment-state detection mechanism has determined the segment to be down.

Here is an example of the information displayed by this command. In this example, the interfaces defined in the command example in Section 2.6.2 are displayed.

```
25:PowerHub:ipx# interface-table
 Port  Network Address        MTU     Encapsulation     State    Cost
 ----  ---------------        ---     -------------     -----    ----
   1         00001001         576        enet            UP       0
   1         55ccdd55         576        802.2           UP       0
   2         55ccdd55         576        802.2           UP       0
   3         55ccdd55         576        802.2           UP       0
   6         00002002         576        enet            UP       1
```

### 2.6.5 Deleting an IPX Interface

The **del-interface** command deletes an IPX interface. Here is the syntax for this command.

```
del-interface <seg-list>|all <network>|all
```

where:

*<seg-list>*|**all** Specifies the segment(s) from which you want to delete the network number. If you specify **all**, the network number is removed from all the PowerHub segments.

*<network>*|**all** Specifies the IPX network you want to delete. If you specify **all**, all IPX networks are deleted from the specified segment(s).

## 2.7 ENABLING IPX FORWARDING

After you define the IPX interfaces (see Section 2.6.2), you are ready to enable IPX forwarding. By enabling IPX forwarding, the IPX software can send and receive RIP and SAP updates, and respond to RIP and SAP requests from stations.

Use the following command to enable IPX forwarding:

```
set|se ipxForwarding|xfw enl|dis
```

where:

**enl|dis** Specifies whether you are enabling or disabling IPX forwarding. The default is **dis**.

## 2.8 USING THE ROUTE TABLE

The PowerHub software stores information about IPX routes in a data structure called the *route table*. This information is used when forwarding IPX packets. The table contains two types of routes:

*Dynamic routes* Directly attached (local) routes and routes learned by the system through IPX RIP. When you add an IPX interface to a PowerHub segment, the software adds a corresponding entry to the IPX route table. Such routes are regarded by the software as local routes. The routing software automatically routes any incoming IPX packet whose destination address is on a directly attached network to the corresponding segment.

Additional information is required to route packets to destinations that are not directly attached. IPX routers use their own version of RIP to dynamically discover routes to non-directly-attached nodes and networks.

*Static routes* Configured using the **add-route** command. You can add static routes for routing to remote IPX networks. Static routes are not subject to the software's aging mechanism and can be removed only by the **del-route** command.

### 2.8.1 Displaying the Route Table

Use the **`route-table`** command to display the IPX route table. Here is the syntax for this command.

```
route-table|rt
   [-c|-r] [-p <seg-list>|-n <network>] [-t]
```

where:

**`-c|-r`** Restricts the display to one of the following:

**`-c`** Only directly connected entries

**`-r`** Only remotely attached entries

**`-p`** *`<seg-list>`* Specifies the segment(s) for which you want to display route information.

**`-n`** *`<network>`* Specifies the IPX network for which you want to display route information.

**`-t`** Displays the total count of UP and DOWN routes.

Here is an example of the display produced by this command:

```
60:PowerHub:ipx# route-table

Destnet   Gway-net  Gway-nodeaddr      Hops  Ticks  State  Age  Ports
--------  --------  -----------------  ----  -----  -----  ---  -----
00001001  --------  ------------         1       2  UP     ---     1
00002002  --------  ------------         1       2  UP     ---     6
55ccdd55  --------  ------------         1       2  UP     ---     1
55ccdd55  --------  ------------         1       2  UP     ---     2
55ccdd55  --------  ------------         1       2  UP     ---     3
008ffff9  96aabb69  00-00-99-88-88-88    2       3  UP     ---     8
054ffff9  f4f4f4f4  00-00-99-22-22-22    2       3  UP     ---     4
064ffff9  f4f4f4f4  00-00-99-22-22-22    2       4  UP     ---     4
011ffff9  96aabb69  00-00-99-11-11-11    2       3  UP     ---     3
165ffff9  00fabcab  00-00-99-44-44-44    2       3  UP     ---     9

Total no. of routes = 10 (10 UP, 0 DOWN)
```

This command displays the following information about IPX routes:

`Destnet` The IPX network number of the destination network.

`Gway-net` If the destination is not directly attached, this field contains the IPX network number of the gateway (IPX router) through which packets for the destination are to be routed.

`Gway-nodeaddr` If the destination is not directly attached, this field contains the node address of the IPX gateway (router) through which packets for the destination are to be routed.

`Hops` The number of gateways, including the PowerHub, that a packet must go through to reach the destination. If a network is directly attached, the hop-count is 1.

`Ticks` The number of 55-mS ticks that can be expected for a packet to reach its destination. If all of the network segments along the route have a bandwidth of 1 Mb/s or more, the number of ticks generally equals the number of hops plus 1. Otherwise, it is larger to account for the slower segments.

`State` This is the state of the route; possible states are `UP` and `DOWN`. When a segment goes down, its state is updated in the interface table. All routes that use this segment are marked `DOWN` in the route table, and all servers that are not accessible except through this segment are marked as `DOWN` in the server table.

When the segment comes back up, its state is again updated in the interface table. All routes that use this segment are marked as `UP` in the route table, and servers that are now accessible through this segment are marked as `UP` in the server table.

`Age` For dynamic routes, the number of seconds that have elapsed since this routing information was received. The `Age` field displays "`---`" for direct/static routes. For RIP entries, the `Age` field displays how long it has been since a routing update for the route has been received.

`Ports` Lists the segments on which packets for this destination should be forwarded.

The software does not contain a command to directly take a static route `DOWN`. To take `DOWN` a static route, use the **`delete-route`** command to remove the route. (See Section 2.8.3.)

### 2.8.2 Adding a Static Route

To assign the route to be used when forwarding to a particular network, use the **`add-route`** command. Here is the syntax for this command.

```
add-route|ar <network> <gw-net> <gw-addr> <seg> <hops>
             <ticks>
```

where:

*`<network>`* Specifies the destination IPX network number, a 32-bit value expressed as up to eight hexadecimal digits. Specify a number in the range from **`1`** through **`ffffffe`**.

*`<gw-net>`* Specifies the network number of the gateway (IPX router) through which packets for the destination network are to be routed. This network number must be the same as a network number that is already configured on the segment specified by the *`<seg>`* argument. Specify a number in the range from **`1`** through **`ffffffe`**.

*`<gw-addr>`* Specifies the IPX node number of the gateway (router) to which packets for the destination network should be forwarded. An IPX node number is actually a 48-bit MAC-layer address. Such an address is expressed in PowerHub commands as six hexadecimal bytes separated by hyphens.

The gateway should be a device connected to a network that is directly attached to the PowerHub segment specified in the *`<seg>`* argument.

*<seg>* Specifies the PowerHub segment on which a packet should be forwarded to reach the specified gateway and, eventually, the specified network.

*<hops>* Specifies the number of hops to the destination, that is, how many gateways a packet must go through to reach the specified network.

A hop-count of **1** corresponds to a direct connection. (Note, however, that you cannot add a route to a network that is directly attached.)

The maximum number of hops is **15**; a hop-count of **16** is synonymous with "infinity" and means that the specified network is unreachable.

*<ticks>* Specifies the typical delay expected for a packet to reach its destination, measured in 55-mS "ticks."

In Ethernet, FDDI, and other networks with bandwidths greater than 1 Mb/s, each network is assumed to create a delay of one tick. If a route includes only such networks, the number of ticks should be set equal to the number of network segments in the route, which is the number of hops plus 1. However, routing paths that include slow, wide-area links (ex: 56 Kb/s leased lines) should have a larger number of ticks to account for the slow links.

Ticks are represented in IPX by 16-bit integers, so the practical maximum number of ticks is far less than the number that you can enter here.

A statically-entered IPX route is always marked as "UP" when it is added. The route is automatically marked as "DOWN" when the corresponding segment is disabled, either manually in the **bridge** subsystem[1] or automatically by the automatic segment-state detection mechanism.

When routing a packet to a remote network, the IPX routing software selects the route with the lowest number of ticks, regardless of whether it is a static route or a dynamic route. When two or more routes to a remote network have an equal number of ticks, the router chooses the route with the smallest number of hops.

An example of the **add-route** command is shown below:

```
50:PowerHub:ip# add-route 008ffff9 96aabb69 0-0-99-88-88-88 8 2 3
Route to 008ffff9 via 96aabb69: added.
```

The result of this command is that packets directed to network 008ffff9 are forwarded on segment 8 to a gateway with address 0-0-99-88-88-88, and can expect to require a total of 2 hops and 3 ticks to reach a station on the destination network.

[1] To manually disable a segment, issue the **bridge port** *<seg-list>* **dis** command, where *<seg-list>* specifies the segment(s) you want to disable. See Section 2.3 in the *PowerHub Software Manual, V 2.6*.

### 2.8.3 Deleting a Static Route

You can completely eliminate a static route using the **`del-route`** command. Here is the syntax for this command.

**`del-route|dr`** *`<network> <gw-net> <gw-addr>`*

where:

*`<network>`* Specifies the destination IPX network number, a 32-bit value expressed as up to eight hexadecimal digits.

*`<gw-net>`* Specifies the network number of the gateway (IPX router).

*`<gw-addr>`* Specifies the IPX node number of the gateway (router).

## 2.9 USING THE ROUTE CACHE

The IPX *route cache* shows, for each segment, the most recently used destination networks. At any time, you can get an at-a-glance picture of IPX routing activity in your network by displaying the IPX route cache.

### 2.9.1 Displaying the Route Cache

Use the **`display-routecache`** command to display the IPX route cache. Here is the syntax for this command.

**`display-routecache|dc [`***`<seg-list>`***`]`**

where:

*`<seg-list>`* Specifies the segments for which you want to display information in the route cache. If you do not specify a segment, information for all segments is shown.

Here is an example of the output produced by this command. The cache displayed in this example is for a PowerHub system containing 14 segments.

```
66:PowerHub:ipx# display-routecache
IPX router cache:
Port 01: empty
Port 02: empty
Port 03: 011ffff9, 96aabb69
Port 04: f4f4f4f4, 054ffff9, 064ffff9
Port 05: empty
Port 06: empty
Port 07: empty
Port 08: 00000022
Port 09: 00fabcab, 165ffff9
Port 10: empty
Port 11: empty
Port 12: empty
Port 13: empty
Port 14: empty
```

**NOTE**: The contents of the route cache can change quite rapidly. As a result, successive **`display-routecache`** commands can give different results.

### 2.9.2 Flushing the Route Cache

The **`flush-routecache`** command removes all entries for all segments from the route cache. After the cache is flushed, new entries are again added, using the cache's "most-recently-used" algorithm. Thus, you can use the **`flush-routecache`** command to ensure that all entries displayed by a subsequent **`display-routecache`** command are fresh.

In the following example, the route cache is flushed once and then quickly displayed two times.

```
67:PowerHub:ipx# flush-routecache
IPX router cache flushed
68:PowerHub:ipx# display-routecache
IPX router cache:
Port 01: empty
Port 02: empty
Port 03: 96aabb69
Port 04: f4f4f4f4
Port 05: empty
Port 06: empty
Port 07: empty
Port 08: empty
Port 09: empty
Port 10: empty
Port 11: empty
Port 12: empty
Port 13: 00001fd1
Port 14: empty

69:PowerHub:ipx# display-routecache
IPX router cache:
Port 01: empty
Port 02: empty
Port 03: 96aabb69, 011ffff9
Port 04: f4f4f4f4, 054ffff9
Port 05: empty
Port 06: empty
Port 07: empty
Port 08: 00000022
Port 09: 00fabcab
Port 10: empty
Port 11: empty
Port 12: empty
Port 13: 00001fd1
Port 14: empty
```

## 2.10 USING THE SERVER TABLE

The PowerHub IPX software stores information about NetWare file servers and other NetWare services in a data structure called the *server table*. The IPX routing software maintains a server table containing information that it uses when advertising services and responding to server information requests using SAP (Service Advertising Protocol). The table contains two types of servers:

*Dynamic servers* Learned by the system through the SAP. IPX file servers, print servers, and other service providers advertise their existence using SAP. This information is learned by all IPX routers in the network. When an IPX station requires a service, it uses SAP to request server information from the nearest router.

*Static servers* Configured by a system administrator, using the **`add-server`** command. The IPX routing software always has SAP enabled, and services are always being discovered and advertised dynamically. Although the information learned through SAP is usually sufficient for good network behavior, there might be occasions in which you would like to make permanent entries in the server table. For example, you can make permanent entries in the server table to ensure quick availability of service information after a network outage. Static service assignments can be used for this purpose.

**NOTE**: Before you can add a server to the PowerHub IPX server table, you must add a route (to the IPX route table) to the server's net. See for Section 2.8.2 information on adding a route.

When responding to IPX stations' requests for the information on the "nearest" server of a given type, the PowerHub IPX software selects the server with the best route as determined from the route table, regardless of whether the server is static (added to the server table permanently by the **`add-server`** command) or dynamic (learned through SAP). If there are equally good routes to two or more servers, the software chooses the server with the least number of hops in the server table.

### 2.10.1 Displaying the Server Table

To display the IPX servers known to the PowerHub system, issue the following command.

**server-table|st**
**[-s** *<s-type>***] [-p** *<seg-list>***] [-n** *<network>***]**
**[-i** *<name>***] [-f | -t | -a]**

where:

**-s** *<s-type>* Specifies the type of service, either a mnemonic or a 16-bit number in the range **0** through **fffe**, expressed as up to four hexadecimal digits:

| **Mnemonics** | **Server-type(hex)** |
|---|---|
| PRINT-QUEUE | 0003 |
| FILE-SERVER | 0004 |
| JOB-SERVER | 0005 |
| PRINT-SERVR | 0007 |
| ARCHIVE-SVR | 0009 |
| REM-BRIDGE | 0024 |
| ADVRT-PRINT | 0047 |

**-p** *<seg-list>* Specifies the segment(s) for which you want to display route information.

**-n** *<network>* Specifies the IPX network number of the server.

**-i** *<name>* If you specify a server name here, only information that applies to the specified server is displayed.

**-f|-t|-a** Specifies the type of entries you want to display:

**-f** Displays the entire server name, up to 48 characters. Otherwise, a maximum of 24 characters is displayed to keep the display within an 80-character line.

**-t** Displays only the total count of UP and DOWN server entries.

**-a** Displays the network number and MAC address of the next-hop gateway.

Here is an example of the output produced by this command.

```
73:PowerHub:ipx# server-table
Server-type Srvr-net Server-node       Sock  Hop  State  Port  Age  Server-name
----------- -------- ----------------- ----  ---  -----  ----  ---  -----------
FILE-SERVER 00fabcab 00-00-88-88-88-88 1010  2    UP     9          eng-server
00ff        f4f4f4f4 00-00-99-66-66-66 f4f4  2    UP     4          corp-server
00fe        f4f4f4f4 00-00-99-66-66-66 f4f4  2    UP     4          boss-toaster
0123        f4f4f4f4 00-00-99-66-66-66 f4f4  2    UP     4          espresso-mkr

Total no. of servers = 4 (4 UP, 0 DN)
```

This command displays the following information from the server table:

`Server-type` Specifies the type of service, either a mnemonic or a 16-bit number in the range **`0`** through **`fffe`**, expressed as up to four hexadecimal digits.

`Srvr-net` The IPX network number of the server.

`Server-node` The IPX node number of the server.

`Sock` The IPX socket number on which the server accepts requests for service.

`Hop` The number of gateways, including the PowerHub system, that a packet must go through to reach the server. If the server is on a directly-attached network, the hop-count is 1.

`State` This is the state of the server; possible states are “UP” and “DOWN.”

`Port` The segment on which the entry was learned.

`Age` For dynamic servers, the number of seconds that have elapsed since this information was received.

`Server-name` The name of the server, up to 48 ASCII characters.

### 2.10.2 Adding a Static Server

To add a server to the server table, use the **`add-server`** command. Here is the syntax for this command.

```
add-server|as
    <s-type> <s-net> <s-addr> <s-sock> <s-hops> <s-name>
```

where:

*`<s-type>`* Specifies the type of service, either a mnemonic or a 16-bit number in the range **`0`** through **`fffe`**, expressed as up to four hexadecimal digits:

| **Mnemonics** | **Server-type(hex)** |
|---|---|
| PRINT-QUEUE | 0003 |
| FILE-SERVER | 0004 |
| JOB-SERVER | 0005 |
| PRINT-SERVR | 0007 |
| ARCHIVE-SVR | 0009 |
| REM-BRIDGE | 0024 |
| ADVRT-PRINT | 0047 |

*<s-net>* Specifies the IPX network on which the server resides, a 32-bit number expressed as up to eight hexadecimal digits.

Note that the PowerHub software does not accept the **`add-server`** command if there is no known route to the server's network at the time the command is given. Specify a number in the range from **`1`** through **`fffffffe`**.

*<s-addr>* Specifies the IPX node number of the server. This is a 48-bit MAC-layer address, expressed as six hexadecimal bytes separated by hyphens.

*<s-sock>* Specifies the IPX socket number on which the specified server accepts requests for service.

*<s-hops>* Specifies the number of hops to the specified server, that is, how many gateways a packet must go through to reach it. The maximum number of hops is **`15`**; a hop-count of **`16`** is synonymous with "infinity" and means that the specified server is unreachable.

*<s-name>* Specifies the name of the server, up to 48 ASCII characters. Server names are case sensitive.

Here is an example of the **`add-server`** command:

```
71:PowerHub:ipx# add-server 4 fabcab 0-0-88-88-88-88 1010 2 eng-server
Server eng-server of type 0004 on net 00fabcab: added.
```

### 2.10.3 Deleting a Static Server

You can completely eliminate a static server assignment using the **`del-server`** command. Here is the syntax for this command.

**`del-server|ds`** *<s-type> <s-name>*

where:

*<s-type>* Specifies the type of service, either a mnemonic or a 16-bit number in the range **`0`** through **`fffe`**, expressed as up to four hexadecimal digits:

| **Mnemonics** | **Server-type(hex)** |
|---|---|
| PRINT-QUEUE | 0003 |
| FILE-SERVER | 0004 |
| JOB-SERVER | 0005 |
| PRINT-SERVR | 0007 |
| ARCHIVE-SVR | 0009 |
| REM-BRIDGE | 0024 |
| ADVRT-PRINT | 0047 |

*<s-name>* Specifies the name of the server, up to 48 ASCII characters. Server names are case-sensitive.

Here is an example of the use of this command.

```
72:PowerHub:ipx# del-server 4 eng-server
Server eng-server of type 0004: deleted from table.
```

## 2.11 CONFIGURING IPX RIP AND SAP PARAMETERS

Earlier sections in this chapter describe how to add static entries to the IPX RIP and SAP tables maintained by the PowerHub software. However, the software contains additional RIP and SAP options you can configure:

- Whether the hub generates updates on a per-segment basis or a per-VLAN basis. (See Section 2.11.1.)
- Whether the hub can generate large (greater than 576 bytes) IPX RIP and SAP packets. (See Section 2.11.5.)
- Talk and listen (send and receive) settings for each interface or segment. (See Section 2.11.)

### 2.11.1 Setting the Control Type

You can set the RIP and SAP control type to change the RIP and SAP update mechanism. Using the **`ripsap-ctrl-type`** command, you can configure the PowerHub software to generate and send a copy of each RIP and SAP packet on a per-VLAN basis instead of on a per-segment basis.

If your IPX configuration does not contain IPX VLANs, PowerHub performance will be the same whether you configure the software to generate updates on a per-segment basis or a per-VLAN basis. In this case, we recommend that you leave the configuration in its default state: generate updates on a per-segment basis.

However, if your configuration does include IPX VLANs, you can enhance performance by configuring the software to use the per-VLAN method for generating the RIP and SAP updates. When you change the control type to **`vlan`**, the software spends less time generating RIP and SAP updates, because it generates only a single update for each network, even if the network spans multiple segments.

To change the RIP and SAP update method, issue the following command:

```
ripsap-ctrl-type|rsct [normal|n vlan|v]
```

where:

**`normal|n`** Specifies that RIP and SAP updates are generated on a per-segment basis. This is the default.

**`vlan|v`** Specifies that RIP and SAP updates are generated on a per-VLAN basis.

If no parameter is used with this command, the current control type is displayed.

**NOTE**: This command affects only IPX RIP and SAP updates. It has no affect on IP RIP updates.

## 2.11.2 Displaying the Configuration

The commands for displaying the talk and listen (send and receive) settings for IPX RIP and SAP differ depending upon the update method used by the software:

- If the update method is per-segment, use the **`rip-showcfg`** and **`sap-showcfg`** commands. (See Section 2.11.2.1.)
- If the update method is per-VLAN, use the **`rip-control-tbl show`** and **`sap-control-tbl show`** commands. (See Section 2.11.2.2.)

### 2.11.2.1 Per-Segment Method

Use the commands in the following sections to display the talk and listen (send and receive) settings for IPX RIP and SAP on the PowerHub segments.

**Displaying the RIP Parameters**

Use the **`rip-showcfg`** command to display the current enabled status of RIP sending and receiving. Here is the syntax for this command.

**`rip-showcfg|rscf [`***`<seg-list>`***`]`**

where:

*`<seg-list>`* Specifies the segments for which you want to display the IPX RIP configuration.

Here is an example of the display produced by this command.

```
91:PowerHub:ipx# rip-showcfg

Port      Talk      Listen
----      ----      ------
   1      yes       yes
   2      yes       yes
   3      yes       yes
   4      yes       yes
   5      yes       yes
   6      yes       yes
   7      yes       yes
   8      yes       yes
   9      no        no
  10      no        no
  11      no        no
  12      no        no
  13      yes       yes
  14      no        no
```

### Displaying the SAP Parameters

Use the **`sap-showcfg`** command to display the current enabled status of SAP sending and receiving. Here is the syntax for this command:

**`sap-showcfg|sscf [`*`<seg-list>`*`]`**

where:

*`<seg-list>`* Specifies the segments for which you want to display the IPX SAP configuration.

Here is an example of the display produced by this command.

```
91:PowerHub:ipx# sap-showcfg

Port     Talk     Listen
----     ----     ------
   1     yes      yes
   2     yes      yes
   3     yes      yes
   4     yes      yes
   5     yes      yes
   6     yes      yes
   7     yes      yes
   8     yes      yes
   9     no       no
  10     no       no
  11     no       no
  12     no       no
  13     yes      yes
  14     no       no
```

### 2.11.2.2 *Per-VLAN Method*

### Displaying the RIP Parameters

Use the **`rip-control-tbl show`** command to display a specified network's RIP control table entries. Here is the syntax for this command:

**`rip-control-tbl show|s [`*`<network>`*`]`**

where:

*`<network>`* Is the network address of the network for which you want to display RIP control table entries. If no network is specified, all RIP control table entries are displayed.

Here is an example of this command:

```
32:PowerHub:ipx# rip-control-tbl show
RIP Update Control Configuration:

Network    Talk    Listen
-------    ----    ------
00000002    no      no
```

**Displaying the SAP Parameters**

Use the **`sap-control-tbl show`** command to display a specified network's SAP control table entries. The syntax for this command is:

**`sap-control-tbl|sco show|s [`***`<network>`***`]`**

where:

| | |
|---|---|
| *`<network>`* | Is the network address of the network for which you want to display SAP control table entries. If no network is specified, all SAP control table entries are displayed. |

## 2.11.3 Setting the Parameters

The PowerHub IPX software advertises and receives IPX routing information using the IPX RIP (Routing Information Protocol). The PowerHub IPX software advertises and receives IPX server information using the SAP (Service Advertising Protocol).

> **NOTE**: The RIP protocol used by IPX is different from RIP used in IP.

The commands for displaying the talk and listen (send and receive) settings for IPX RIP and SAP differ depending upon the update method used by the software:

- If the update method is per-segment, use the **`rip-pset`** and **`sap-pset`** commands. (See Section 2.11.2.1.)
- If the update method is per-VLAN, use the **`rip-control-tbl add`** and **`sap-control-tbl add`** commands. (See Section 2.11.2.2.)

### *2.11.3.1 Per-Segment Method*

**Setting the RIP Parameters**

To enable IPX RIP sending (**`talk`**) or receiving (**`listen`**), use the **`rip-pset`** command. Here is the syntax for this command:

**`rip-pset|rpse`** *`<seg-list>`***`|all`** *`<parm-list>`***`|all y|yes|n|no`**

where:

*`<seg-list>`***`|all`** Specifies the segments for which you are setting IPX RIP sending or receiving. You can specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments. If you specify **`all`**, IPX RIP is enabled or disabled for all segments.

*`<parm-list>`***`|all`** Specifies whether you are setting the software to receive RIP updates, send them, or both. Valid options are:

| | |
|---|---|
| **`talk|ta`** | Enables or disables RIP sending. |
| **`listen|li`** | Enables or disables RIP receiving. |

If you specify **`all`**, the enabled status of both RIP sending and RIP receiving is set. If you specify the options individually, separate them with commas.

**`y|yes|n|no`** Specifies whether you are enabling (**`yes`**) or disabling (**`no`**) sending or receiving. The default is **`yes`** for **`talk`** and **`listen`**.

### Setting the SAP Parameters

To enable SAP sending (**`talk`**) or receiving (**`listen`**), use the **`sap-pset`** command. Here is the syntax for this command:

**`sap-pset|spse`** *`<seg-list>`***`|all`** *`<parm-list>`***`|all y|yes|n|no`**

where:

*`<seg-list>`***`|all`** Specifies the segments for which you are setting the SAP sending or receiving. You can specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments. If you specify **`all`**, SAP is enabled or disabled for ***all*** segments.

*`<parm-list>`***`|all`** Specifies whether you are setting the PowerHub software to receive SAP updates, send them, or both. Valid options are:

**`talk|ta`** Enables or disables SAP sending.

**`listen|li`** Enables or disables SAP receiving.

If you specify **`all`**, the enabled status of both SAP sending and SAP receiving is set. If you specify the options individually, separate them with commas.

**`y|yes|n|no`** Specifies whether you are enabling (**`yes`**) or disabling (**`no`**) the sending or receiving of SAP updates. The default is **`yes`** (enabled) for **`talk`** and **`listen`**.

### *2.11.3.2 Per-VLAN Method*

### Setting the RIP Parameters

Use the **`rip-control-tbl add`** command to set one or more RIP parameters for a network address and add the information to the RIP update control table. The syntax for this command is:

**`rip-control-tbl|rco add|a`** *`<network>`*

  *`<parm-list>`***`|all yes|y|no|n`**

where:

*`<network>`* Specifies the network address of the network for which you want to enable or disable RIP talk or listen.

*`<parm-list>`***`|all`** Specifies either a comma-separated list of parameters or **`all`** for all parameters. Select the following parameters for a network address:

**`talk|ta`** Enables or disables the sending of RIP update packets to the specified network.

**`listen|li`** Enables or disables the learning of routes from RIP packets received from the specified network.

If you specify **`all`**, all parameters are set to either **`yes`** or **`no`**.

**`yes|y|no|n`** Specifies whether you are enabling (**`yes`**) or disabling (**`no`**) sending or receiving.

In the example that follows, RIP packet sending and receiving has been enabled on IPX network 2002.

```
37:PowerHub:ipx# rip-control-tbl add 00002002 talk,listen yes
```

**Setting the SAP Parameters**

Use the **`sap-control-tbl add`** command to set one or more SAP parameters for a network address and add the information to the SAP update control table. The syntax for this command is:

```
 sap-control-tbl|sco add|a
    <network> <parm-list>|all yes|y|no|n]
```

where:

*`<parm-list>`*`|all` Specifies either a comma-separated list of parameters or **`all`** for all parameters. Select the following parameters for a network address:

- **`talk|ta`** Enables or disables the sending of SAP update packets to the specified network.
- **`listen|li`** Enables or disables the learning of routes from SAP packets received from the specified network.

If the *`<param-list>`* is not entered, all parameters are set to **`no`**.

**`yes|y|no|n`** Specifies whether you are enabling (**`yes`**) or disabling (**`no`**) sending or receiving.

## *2.11.4 Removing Parameter Settings*

As is true for displaying or adding RIP and SAP parameters, the commands you use to change the parameters depends upon the update method the software is using.

### *2.11.4.1 Per-Segment Method*

To change the settings for a RIP or SAP network, use the **`rip-pset`** or **`sap-pset`** command to enter the new settings for the segment. (See Section 2.11.3.1.)

### *2.11.4.2 Per-VLAN Method*

To change the settings for a RIP or SAP network, delete the existing settings, then re-add them using the **`rip-control-tbl add`** or **`sap-control-tbl add`** command. (See Section 2.11.3.2 for descriptions of the commands for adding new settings.)

**Deleting RIP Parameters**

Use the **`rip-control-tbl del`** command to delete a specific network's RIP control table entries. The syntax for this command is:

```
rip-control-tbl|rco del|d <network>
```

where:

*`<network>`* Is the network address of the network for which you want to delete RIP control table entries.

**Deleting SAP Parameters**

Use the **`sap-control-tbl del`** command to delete a specific network's RIP control table entries. The syntax for this command is:

```
sap-control-tbl|sco del|d <network>
```

where:

*`<network>`* Is the network address of the network for which you want to delete SAP control table entries.

### 2.11.5 Enabling Large Packets

In software version 2.6, IPX RIP and IPX SAP packets larger than 576 bytes (the default minimum) can be generated. To change the default, use the **`set large-rip-sap-pkt`** command to enable the software to generate large RIP and SAP packets.

The syntax for this command is:

```
set|se large-rip-sap-pkt|lpkt enable|enl disable|dis
```

where:

**`enable|enl|disable|dis`**
Specifies that the PowerHub software generate or not generate IPX RIP or IPX SAP packets larger than 576 bytes.

> **NOTE**: The MTU setting for the IPX interfaces you define on the hub needs to be more than 576 bytes to generate larger RIP and SAP packets. To adjust the MTU setting, see Section 2.6.2 for information on the **`add-interface`** command.

## 2.12 CONFIGURING IPX TYPE-20 FORWARDING

IPX type-20 packets are broadcast packets and, by default, are forwarded to or received on all PowerHub segments.

### 2.12.1 Displaying the Enabled State of Type-20 Forwarding

You can display the setting for type-20 forwarding for the entire hub or for specific segments. The following sections describe the commands you use to display and change the enabled state for type-20 forwarding.

> **NOTE**: Type-20 forwarding must be enabled for the entire hub for forwarding to work on individual segments.

### *2.12.1.1 For the Entire Hub*

To display the setting for the entire hub, issue the **`showcfg`** command. (See Section 2.5 for information.)

### *2.12.1.2 For Individual Segments*

To display the state for individual segments, issue the following command:

**`type20-port-forwarding|tpfw show`** *`<seg-list>`***`|all`**

where:

*`<seg-list>`***`|all`** Specifies a segment, a comma-separated list of segments, or a hyphen-separated range of segments for which you want to display the type-20 packet forwarding state.

If you specify **`all`**, the type-20 forwarding state of all segments is displayed.

Here is an example of this command:

```
15:PowerHub:ipx# type20-port-forwarding show 1,4,6,8

Type 20 Packet Forwarding Configuration of Ports:
    Port            Status
    ----            ------
       1            enabled
       4            enabled
       6            enabled
       8            enabled
```

## *2.12.2 Changing the Enabled State of Type-20 Forwarding*

You can enable or disable type-20 forwarding for individual segments, or for the entire hub. Forwarding is enabled by default for the entire hub. To disable forwarding on segments, but enable it on others, leave forwarding enabled for the hub and disable it for specific segments.

### *2.12.2.1 For the Entire Hub*

To enable or disable the forwarding of type-20 packets for the entire hub, issue the following command:

**`set type20-forwarding|t20fw enable|enl disable|dis`**

#### 2.12.2.2 For Individual Segments

Use the **`type20-port-forwarding`** command to show whether type-20 packet forwarding is enabled or disabled on specific segments. The syntax for this command is:

```
type20-port-forwarding|tpfw enl|dis <seg-list>|all
```

where:

**`enl|dis`** Specifies whether you are enabling or disabling type-20 packet forwarding. The default is **`enl`** (enabled).

*`<seg-list>`*`|`**`all`** Specifies a segment, a comma-separated list of segments, or a hyphen-separated range of segments for which you want to enable or disable type-20 packet forwarding.

If you specify **`all`**, the type-20-forwarding state of all segments is set.

### 2.12.3 Adjusting the Broadcast Interval and Aging Timers

You can adjust the following RIP and SAP timers in the PowerHub software.

*Broadcast interval* The PowerHub IPX software generates and transmits RIP and SAP updates at regular intervals. The RIP updates contain information about the IPX routes known to the PowerHub software. The SAP updates contain information about the IPX servers known to the PowerHub software.

The default interval for RIP and SAP updates is 60 seconds. Every 60 seconds, the hub generates and transmits IPX RIP and SAP updates. Depending on whether you configured RIP and SAP updates to use the per-segment method or the per-VLAN method, updates are generated for each segment, or for each network.

*Aging timer* *Aging* is a mechanism that periodically clears learned entries from the RIP and SAP tables. At an interval you specify (the *aging interval*) the PowerHub software determines which of the learned entries in the table have not been recently used. For proper RIP and SAP reporting, the aging interval must be at least three times the duration of the broadcast interval. If an entry is not used during the specified interval, it is discarded.

A separate broadcast interval and aging timer are maintained for IPX RIP and for IPX SAP. The following sections describe how to change these parameters for RIP and SAP.

#### 2.12.3.1 RIP

Use the **set rip-timer** command to set the RIP age timer and the RIP broadcast interval. Here is the syntax for this command:

**set|se rip-timer|rtmr** *<rip-bcast-intvl>* **[***<rip-age>***]**

where:

*<rip-bcast-intvl>* Sets the RIP broadcast interval. Specify a value from **60** to **600** seconds. The default is **60** seconds.

*<rip-age>* Sets the RIP age timer. If specified, the RIP age timer value must be at least three times the value of the RIP broadcast interval. Specify a value between **180** and **1800** seconds. If unspecified, this argument defaults to three times the value of the RIP broadcast interval.

Here is an example of this command:

```
22:PowerHub:ipx# set rip-timer 100 300
```

#### 2.12.3.2 SAP

Use the **set sap-timer** command to set the SAP age timer and the SAP broadcast interval.

Here is the syntax for this command:

**set|se sap-timer|stmr** *<sap-bcast-intvl>* **[***<sap-age>***]**

where:

*<sap-bcast-intvl>* Sets the SAP broadcast interval. Specify a value from **60** to **600** seconds. The default is **60** seconds.

*<sap-age>* Sets the SAP age timer. If specified, the SAP age timer value must be at least three times the value of the SAP broadcast interval. Specify a value between **180** and **1800** seconds. If unspecified, this argument defaults to three times the value of the SAP broadcast interval.

## 2.13 DISPLAYING AND CLEARING STATISTICS

The **ipx** subsystem maintains statistics on:

- IPX packets and type-20 (NetBIOS) packets.
- RIP updates.
- SAP updates.

### 2.13.1 Displaying IPX and Type-20 Statistics

Use the **stats** command to display IPX or type-20 packet statistics. Here is the syntax for this command:

**stats|s [ipx|type20] [-t]**

where:

**ipx|type20** Specifies the type of packet for which you want to display statistics:

- **ipx** IPX packet statistics. This is the default.
- **type20** NETBIOS packet statistics.

**-t** Optionally displays statistics collected since the most recent system reset, rather than those collected since the most recent clear. (Statistics are cleared using the **stats-clear** command.)

Here is an example of the output produced by the **stats ipx** command.

```
80:PowerHub:ip# stats ipx
IPX statistics: count since last stats clear
Datagrams received:                 2302091
Header errors received:             0
Address errors received:            0
Datagrams forwarded:                2302091
Unknown protocols received:         0
Incoming datagrams discarded:       0
Datagrams delivered to higher layer: 2258
Datagrams sent:                     6658
```

Here is an example of the output produced by the **stats type20** command.

```
81:PowerHub:ip# stats type20
Type-20 statistics: count since last stats clear
Packets   received:                 0
Packets   forwarded:                0
Packets   discarded:                0
Packets   in error:                 0
```

Here is an example of the use of the **-t** argument with the **stats** command. In this example, IPX statistics collected since the last system reset are displayed.

```
83:PowerHub:ip# stats ipx -t
IPX statistics: Total count since last system reset
Datagrams received:                 2305309
Header errors received:             0
Address errors received:            0
Datagrams forwarded:                2305309
Unknown protocols received:         0
Incoming datagrams discarded:       0
Datagrams delivered to higher layer: 2261
Datagrams sent:                     6664
```

#### 2.13.1.1 Clearing IPX and Type-20 Statistics

To clear statistics, use the **`stats-clear`** command. Here is the syntax for this command.

```
stats-clear|sc [ipx|type20]
```

where:

**`ipx|type20`** Specifies the type of packet for which you want to clear statistics:

| | |
|---|---|
| **`ipx`** | IPX packet statistics |
| **`type20`** | NETBIOS packet statistics |

If you do not specify a packet type, statistics for both IPX and type-20 packets are cleared.

### 2.13.2 Displaying RIP Statistics

Use the **`rip-stats`** command to display RIP statistics for IPX. Here is the syntax for this command:

```
rip-stats|rst [-t]
```

where:

**`-t`** Optionally displays statistics collected since the most recent system reset, rather than those collected since the most recent clear. (Statistics are cleared using the **`rip-stats-clear`** command.)

#### 2.13.2.1 Clearing RIP Statistics

To clear RIP statistics, use the **`rip-stats-clear`** command. Here is the syntax for this command:

```
rip-stats-clear|rstc
```

### 2.13.3 Displaying SAP Statistics

Use the **`sap-stats`** command to display SAP statistics for IPX. Here is the syntax for this command:

```
sap-stats|sst [-t]
```

where:

**`-t`** Optionally displays statistics collected since the most recent system reset, rather than those collected since the most recent clear. (Statistics are cleared using the **`sap-stats-clear`** command.)

Here is an example of the display produced by the **sap-stats** command.

```
95:PowerHub:ipx# sap-stats
SAP statistics: count since last stats clear
Packets received:           142
Packets sent:               352
Nearest Requests received:  0
Requests received:          0
Responses received:         142
Requests sent:              0
Nearest Responses sent:     0
Responses sent:             352
```

#### 2.13.3.1 Clearing SAP Statistics

To clear SAP statistics, use the **sap-stats-clear** command. Here is the syntax for this command:

**sap-stats-clear|sstc**

## 2.14 USING IPX HELPER

This section describes how to use the IPX Helper feature. *IPX Helper* lets the PowerHub system forward unknown IPX broadcast packets, which normally would be dropped, onto specified networks. This feature forwards the unknown IPX broadcast packets without using the IPX SAP protocol.

When you assign an IPX helper address to a segment, and an unknown IPX broadcast packet with the specified destination socket number is received on that segment:

- The IPX broadcast packet destination network number and destination node address are replaced with the number and address specified in the **helper add** command. (See Section 2.14.2.)
- The IPX broadcast packet then is forwarded onto all other segments.

### 2.14.1 Enabling IPX Helper

To use IPX Helper, you first must enable it by issuing the following command:

**set|se helper enable|enl disable|dis**

where:

**enable|enl|disable|dis**

Specifies whether you are enabling or disabling the IPX Helper feature.

### 2.14.2 Adding an IPX Helper Address

Use the **helper add** command to add an IPX helper address to a segment. Here is the syntax for this command:

**helper add** *<network> <node-addr> <socket> <seg-list>*

where:

*<network>* Specifies a network number or the value **ffffffff** to specify all net broadcast.

*<node-addr>* Specifies the unicast address or the broadcast address **ff-ff-ff-ff-ff**.

*<socket>* Specifies a socket number in hexadecimal notation. To specify any socket number, enter the value **ffff**.

*<seg-list>* Specifies a segment, a comma-separated list of segments, or a hyphen-separated range of segments.

Here is an example of how to add an IPX Helper address. In this example, a broadcast address is defined.

```
95:PowerHub:ipx# helper add aabbccdd ff-ff-ff-ff-ff-ff ffff 1
```

### 2.14.3 Displaying an IPX Helper Address

Use the **helper show** command to display IPX helper addresses assigned for all segments. Here is an example of the information displayed by this command. In this example, the IPX Helper address configured in the example in Section 2.14.2 is displayed.

```
220:PowerHub:ipx# helper show

PORT   NETWORK    NODE ADDRESS          SOCKET NUMBER
----   -------    ------------          -------------
   1  aabbccdd    ff-ff-ff-ff-ff-ff     ffff
```

### 2.14.4 Deleting an IPX Helper Address

Use the **helper delete** command to delete an IPX helper address assigned to a segment. The syntax for this command is:

**helper delete** *<seg-list>*

where:

*<seg-list>* Specifies a segment, a comma-separated list of segments, or a hyphen-separated range of segments.

# 3 DECnet Commands

The PowerHub Intelligent Switching Hub contains a complete set of DECnet Phase IV routing software for use in DECnet networks. The routing engine works side-by-side with the MAC-layer bridging software. With appropriate configuration, the PowerHub system can be set up to perform DECnet routing on any segments.

This chapter assumes that you are familiar with the basic requirements of DECnet networks and the DECnet protocol. For further information on this subject, refer to a DECnet guide, such as the *DECnet Phase IV General Description,* Order No. AA-N149A-TC, (Digital Equipment Corporation, 1982).

This chapter describes the commands and facilities of the DECnet subsystem. To set up the PowerHub system for DECnet routing, you must perform the following steps:

(1) Allocate memory for DECnet routing. (See Section 3.2.1.)

(2) Assign the DECnet node ID using the **`set-node-param node-id`** command. (See Section 3.3.)

(3) If the PowerHub system is to be a Level-2 router, select it with the **`set-node-param node-type`** command. (See Section 3.3.)

(4) Turn on DECnet routing with the **`set-node-param dec-forwarding enable`** command. (See Section 3.3.)

(5) Enable DECnet routing on the desired segments with the **`set-port-param`** command. (See Section 3.4.)

A large number of nodes may necessitate increasing the maximum limits for these parameters with the **`set-node-param max-area-num`** and **`set-node-param max-node-num`** commands. See Section 3.3.2 on page 70 for further details on this topic.

After setting up DECnet routing, you can check connectivity to hosts and other routers using the PowerHub software's **`display`** and **`statistics`** commands.
(See Section 3.5.1 on page 76 and Section 3.6 on page 81.)

After you are satisfied with your DECnet configuration, we recommend that you save your configuration using the **`mgmt savecfg`** *`<file-name>`* command or the **`tftp svcfg`** *`<file-name>`* command. See the *Installation and Configuration Manual* for your PowerHub system for information.

## 3.1 ACCESSING THE DECNET SUBSYSTEM

To access the **dec** subsystem, issue the following command from the runtime command prompt:

```
dec
```

Most of the commands in this chapter assume that you have changed the focus of the command prompt to “`dec`.” A few commands, such as **`getmem`**, are not in the **`dec`** subsystem. This chapter identifies such commands by listing their subsystem name with the command (ex: **`main getmem`**.)

## 3.2 DECNET SUBSYSTEM COMMANDS

Table 3–1 lists and describes the **dec** subsystem commands and their syntax. For each command, the management capability (root or monitor) is listed, as well as the section that contains additional information about the command.

**TABLE 3–1** DEC subsystem commands.

| Command and Description | Capability* | See… |
|---|---|---|
| **`display-area-tbl|dat [`***`<area>`***`]`**<br>Display routes to all the reachable areas. | R or M | 3.5.4 |
| **`display-endnode-adj|dea [`***`<node>`***`]`**<br>Displays adjacencies to all the end nodes. | R or M | 3.5.3 |
| **`display-node-param|dnp`**<br>Displays DECnet node parameters. | R or M | 3.5.5 |
| **`display-port-param|dpp [`***`<seg-list>`***`|all]`**<br>Displays DECnet segment parameters. | R or M | 3.4.2 |
| **`display-route-tbl|drt [`***`<node>`***`|`***`<area-rtr>`***`]`**<br>Displays routes to all reachable nodes. | R or M | 3.5.4 |
| **`display-router-adj|dra [`***`<node>`***`]`**<br>Displays adjacencies to all the routers. | R or M | 3.5.2 |
| **`display-routecache|dc [`***`<seg-list>`***`|all]`**<br>Displays DECnet route cache. | R or M | 3.5.5 |
| **`flush-routecache|fc`**<br>Flushes DECnet route cache. | R | 3.5.5 |
| ***R= Root, M=Monitor. | | |

**TABLE 3–1** (Continued) DEC subsystem commands.

| Command and Description | Capability* | See… |
|---|---|---|
| **node-stats\|ns [-t]**<br>Displays DECnet node statistics. | R or M | 3.6 |
| **node-stats-clear\|nsc**<br>Clears DECnet node statistics. | R | 3.6 |
| **port-stats\|ps** *<seg-list>***\|all [-t]**<br>Displays DECnet packet statistics for the specified segments. | R or M | 3.6 |
| **port-stats-clear\|psc [***<seg-list>***\|all]**<br>Clears DECnet packet statistics for the specified segments. | R | 3.6 |
| **routing-status\|rs**<br>Shows status of DECnet forwarding and the routing state of all the segments. | R or M | 3.4.2 |
| **set-node-param\|snp** *<node-param> <node-info>*<br>Configures the various DECnet node parameters. | R | 3.3.2 |
| **set-port-param\|spp**<br>*<seg-list>***\|all** *<seg-param> <seg-info>*<br>Shows status of DECnet forwarding and the routing state of all the segments. | R | 3.4.1 |

***R= Root, M=Monitor.

### 3.2.1 Allocating Memory

Before you begin using the **dec** subsystem, allocate memory for the subsystem by issuing the **getmem dec** command (located in the **main** subsystem), as shown in the following example:

```
1:PowerHub:main# getmem dec
Memory allocated for DEC routing.
2:PowerHub:main#
```

If memory has been allocated for DECnet routing at the time you save the configuration with a **mgmt savecfg** command or **tftp savecfg** command, the corresponding **getmem dec** command is placed in the configuration file ahead of the other DECnet configuration commands. Thus, you only need to type the **getmem** command when you first configure the PowerHub system for DECnet routing.

**NOTE**: FORE Systems recommends that you allocate memory for the IPX subsystem immediately after you boot the PowerHub system to ensure that the memory you request is available. For more information, he *PowerHub Installation and Configuration Manual* for your PowerHub system.

You cannot deallocate memory. To free allocated memory, make sure the configuration file does not contain a **`main getmem`** command, then reboot the software.

You can verify that memory has been allocated using the **`dec rs`** command. If memory has not been allocated, you are not allowed to execute the command.

```
3:PowerHub:main# dec rs
DECnet routing status:

Node/Port               Management State   Routing State
---------               ----------------   -------------
DEC-Forwarding          Disabled           Down

Port  1                 Disabled           Down
Port  2                 Disabled           Down
<additional rows omitted for brevity>
4:PowerHub:main#
```

## 3.3 NODE CONFIGURATION

When placed in a DECnet internetwork, the PowerHub system acts as a standard router, capable of connecting many different DECnet networks together. It determines the identity and location of its neighbors through standard DECnet Phase IV protocols, and finds the closest path to each. It then uses this information to route packets that arrive at the input segments.

The DECnet Phase IV routing protocol calls for each node to have an “area number” (between 1 and 63), as well as a node ID (from 1 to 1023). For Level-1 networks (consisting only of Level-1 endnodes and routers), the area numbers are identical and unused. In Level-2 networks, an “area” is defined as a collection of several nodes with identical area numbers. These areas are connected by Level-2 routers. If the PowerHub system is configured as a Level-2 router (with the **`set-node-param node-type area-router`** command), it uses an extended set of routing protocols that can connect nodes from different areas. Normal nodes can only route packets directly to other nodes within their area (those with matching area numbers). If they are called upon to send a packet to a node in another area, they send it to the “nearest” Level-2 router. This Level-2 router keeps track of routes to all other Level-2 routers, as well as routes to normal nodes within its area. This two-level hierarchy allows for a larger network with manageable routing tables.

When the PowerHub system is configured as a Level-2 router, it locates all nodes in its area *and* all other Level-2 routers. Note that this places some restrictions on the topology of the network, described below. The PowerHub system announces itself as a Level-2 router to the normal nodes in its area so that all inter-area packets are sent through it (thus a pair of Level-2 routers are needed for inter-area packets: one in each area). If you place the PowerHub system into a network that uses Level-2 routing, and you wish the PowerHub system to serve as a Level-2 router, be sure to turn this option on (with **`set-node-param node-type area-router`**). If you do not want the PowerHub system to be a Level-2 router, or if your network uses only Level-1 routing, turn this option off (with **`set-node-param node-type router`**). The default is to use only Level-1 routing.

The use of areas in DECnet Level-2 routing places some restrictions upon the topology of these networks:

- Each node must be able to get to each other node in its area without the use of Level-2 routers and without leaving its area. Consequently, all the nodes in a given area must form a contiguous group. If all nodes from other areas are removed, leaving only the nodes from this area, there can be no isolated nodes remaining. This restriction also applies to Level-2 router nodes.
- The set of all Level-2 router nodes must form a contiguous group so that any packet going from one Level-2 router to another can travel only through other Level-2 router nodes.
- There can not be multiple links between adjacent routers. If two routers are directly connected by more than one segment, the DECnet protocol must be enabled and running on only one of those links. Failure to ensure DECnet is running on only one link results in changes to the routing table every time the doubly-connected nodes discover each other. Such a double connection causes the routing table to be continually flushed, resulting in poor performance and unreachable nodes.

  This situation is represented graphically below:

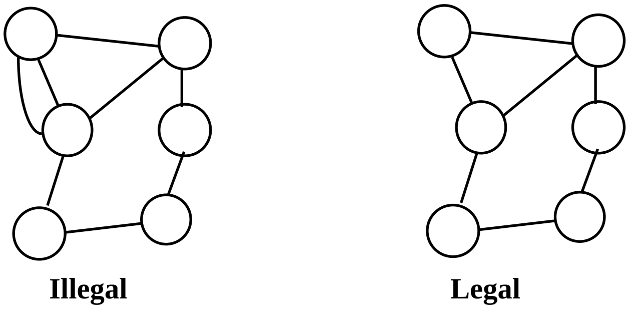


**FIGURE 3–1** Level-2 router nodes.

There is also a topological consideration that improves the efficiency of DECnet Level-2 networks. When a heavily populated broadcast medium is used, such as an Ethernet segment with several nodes, all the nodes on the same segment should be assigned the same area number. The reason is that two nodes with different area numbers must use Level-2 routing to communicate. Therefore this cable segment must have a pair of Level-2 routers on it (one for each area), and the communication path requires three hops, even though the nodes are on the same segment and could communicate directly by other protocols. To avoid these extra hops, all nodes that can communicate directly with each other should be placed in the same area by giving them identical area numbers.

### *3.3.1 DECnet Network Topology Restrictions*

- All nodes in a given area must be connected.
- All Level-2 nodes must be connected.
- No redundant paths are allowed between adjacent routers.

Note that these restrictions do not prevent the same network segment from serving both Level-1 nodes and Level-2 nodes. Thus the same Ethernet segment can serve to connect Level-1 routers, Level-1 endnodes, and Level-2 routers. The requirement is that all of an area be contiguous; nodes from different areas can be on the same segment as long as data moving within one area does not have to pass through the other area’s nodes in order to reach its destination.

### *3.3.2 The Set-Node-Param Commands*

Use the **`set-node-param`** command to change several parameters that affect the entire PowerHub system (as opposed to **`set-port-param`**, which affects individual segments). Most of these have reasonable default values, so they should only be changed if you have unusual requirements. Several of these control the size of internal routing tables. If you have a large network, they may need to be raised, but otherwise they should be kept low to conserve memory. A list of these parameters is contained in the help message for the **`set-node-param`** command. The descriptions of the **`set-node-param`** parameters are included in their help listings in Table 3–2 on page 70.

**TABLE 3–2 `set-node-param`** options.

```
set-node-param|snp dec-forwarding|dfw enl|dis
    Enable or disable routing of DECnet packets.

set-node-param|snp max-adj-endnodes|mae <value>
    Set the number of endnode adjacencies supported by this router.
    Range for <value>: 1 - 1023

set-node-param|snp max-adj-routers|mar <value>
    Set the number of broadcast router adjacencies supported by this router.
    Range for <value>: 1 - 128

set-node-param|snp max-area-num|man <value>
    Set the maximum area number allowed in the entire network.
    Range for <value>: 1 - 63
    NOTE: <value> must be greater than or equal to
          maximum area number in use.

set-node-param|snp max-cost-to-area|mca <value>
    Set the maximum cost possible in a path to a reachable area.
    Range for <value>: 1 - 1022
    NOTE: <value> must be greater than or equal to
          (actual max hops to an area * 25).
```

**TABLE 3–2 (Continued) `set-node-param`** options.

```
set-node-param|snp max-cost-to-node|mcn <value>
    Set the maximum cost possible in a path to a reachable node.
    Range for <value>: 1 - 1022
    NOTE: <value> must be greater than or equal to
          (actual max hops in area * 25).

set-node-param|snp max-hops-to-area|mha <value>
    Set the maximum hops possible in a path to a reachable area.
    Range for <value>: 1 - 30
    NOTE: <value> must be greater than or equal to
          actual max hops to any area.

set-node-param|snp max-hops-to-node|mhn <value>
    Set the maximum hops possible in a path to a reachable node.
    Range for <value>: 1 - 30
    NOTE: <value> must be greater than or equal to
          actual max hops in an area.

set-node-param|snp max-node-num|mnn <value>
    Set the maximum node number allowed within this area.
    Range for <value>: 1 - 1023
    NOTE: <value> must be greater than or equal to
          maximum node number in use.

set-node-param|snp max-visits|mvs <value>
    Set the maximum visits for a packet before the router assumes that
    the packet is looping.
    Range for <value>: 31 - 60
    NOTE: <value> must be greater than equal to the actual maximum path
          in the entire network.

set-node-param|snp node-id|nid <area>.<node>
    Set the node identifier for this router.
    Range for <area>: 1 - 63
    Range for <node>: 1 - 1023

set-node-param|snp node-type|ntp router|rt|area-router|ar
    Set the type of routing supported by this router. The options are:
    router|rt: level 1 routing,  area-router|ar: level 2 routing.

set-node-param|snp update-time|upt <seconds>
    Set background timer for sending routing updates.
    Range for <seconds>: 1 - 1200
```

### 3.3.3 Configuring the Hub as a DECnet Node

First, set the maximum node number used in this area. To do this, use the **`max-node-num`** parameter:

**`set-node-param|snp max-node-num|mnn`** *<value>*

This determines the number of nodes that can exist within the PowerHub system's area. The routing software ignores any packets from nodes outside this range. The default is 255, so you must increase it if you have nodes with larger numbers. The DECnet protocol requires node numbers to be in the range 1 to 1023, so you cannot raise the **`max-node-num`** parameter above 1023.

```
171:PowerHub:dec# snp mnn 1023
Okay
```

Next, you must assign the PowerHub system's node ID. Use the **node-id** parameter:

**set-node-param|snp node-id|nid** *<area>*.*<node>*

This command instructs the PowerHub system to use the specified address for all DECnet communications. The *<area>* parameter must match the area in which the PowerHub system has been placed; recall that the DECnet definition of "area" is the set of nodes that have the same area numbers. The *<node>* parameter can be any value that is unique among all nodes in the specified area.

```
172:PowerHub:dec# snp nid 5.1023
Okay
```

Select the type of routing that needs to be done by this router. Use the **node-type** parameter command:

**set-node-param|snp node-type|ntp router|rt | area-router|ar**

This command determines what kind of routing the PowerHub system performs. If you choose "**router**," the PowerHub system performs only Level-1 routing. A Level-1 router keeps track of nodes within its own area only, and does not try to determine routes to other areas. If it receives data for another area, it sends it to the nearest Level-2 router. The PowerHub system acts as a Level-1 router by default.

If you choose "**area-router**," the PowerHub system also performs Level-2 routing. Level-2 is a superset of Level-1: the node routes data to nodes within its area, as well as find routes to other areas. All Level-2 routers find all other Level-2 routers (including those in other areas), and inter-area traffic is sent to a distant Level-2 router for local distribution. If you configure the PowerHub system as a Level-2 router, be aware of the topological restrictions listed in Section 3.3.

By default, the router performs only Level-1 routing. No changes need to be made to this parameter if the PowerHub system is going to be used as a Level-1 router.

For Level-2 routing, type: **set-node-param node-type area-router**.

```
173:PowerHub:dec# snp ntp area-router
Okay
```

Activate DECnet routing with the **set-node-param dec-forwarding enl** command:

**set-node-param|snp dec-forwarding|dfw enl|dis**

This is the primary command which turns on all of the DECnet routing software. However, to have a useful configuration, you must still specify two or more segments that use DECnet. This is accomplished with the **set-port-param** *<seg-list>* **mgmt-state enl** command, described in Section 3.4.

```
191:PowerHub:dec# snp dfw enl
Okay
```

Verify the node configuration with the **`display-node-param`** command.

```
195:PowerHub:dec# display-node-param
DECnet node configuration
-------------------------
DEC-forwarding:     Enabled
Max-Area-Num:       63
Max-Node-Num:       1023
Max-Adj-Endnodes:   1023
Max-Adj-Routers:    128
Max-Cost-To-Area:   100
Max-Hops-To-Area:   16
Max-Cost-To-Node:   125
Max-Hops-To-Node:   30
Max-Visits:         60
Node-Type:          Area Rtr
Node-ID:            5.1023
Routing-State:      Up
Update-Time:        60 seconds
```

Now configure one or more segments to use DECnet forwarding. See Section 3.4.2 on page 74 for this procedure.

## 3.4 SEGMENT CONFIGURATION

Once the PowerHub system is configured to forward DECnet packets, you must designate one or more segments as DECnet segments to make the software interpret and forward the correct packets. This step also causes the software to transmit and accept routing control packets over these segments, enabling it to discover neighboring endnodes and routers. There are also several parameters associated with each segment that can be set to tune network performance.

### 3.4.1 The Set-Port-Parameter Commands

The **`set-port-parameter`** command can configure several segment-related parameters which affect system performance. The most notable is the cost parameter, which can be set to encourage the use of one link over another. For example, if you set the cost lower on a high-speed FDDI link than on a low-speed Ethernet link, packets are sent over FDDI rather than Ethernet. The default cost for all segments is 10. This might have to be changed to match cost values on other DECnet nodes in the network. The parameters that can be set with this command are included in their help listings, shown in Table 3–2 on page 70.

**TABLE 3–3 set-port-parameter** options.

```
set-port-parameter|spp <port-list> block-size|bsz <size>
    Set the data link block size for the port(s) in <port-list>.
    <port-list> is a comma-separated list of ports or "all".
    Range for <size>: 30 - 1498

set-port-parameter|spp <port-list>  cost|cos <value>
    Set the cost for the ports in <port-list>.
    <port-list> is a comma-separated list of ports or "all".
    Range for <value>: 1- 1022

set-port-parameter|spp <port-list> hello-time|htm <time>
    Set the interval for sending hello packets on the port(s) in
    <port-list>.
    <port-list> is a comma-separated list of ports or "all".
    Range for <value>: 1 - 8191

set-port-parameter|spp <port-list> max-routers|mrt <value>
    Set the number of broadcast router adjacencies supported on the port(s)
    in <port-list>.
    <port-list> is a comma-separated list of ports or "all".
    Range for <value>: 1- 10

set-port-parameter|spp <port-list> mgmt-state|mst enl|dis
    Enable/disable receiving and sending DECnet packets on the
    port(s) in <port-list>
    <port-list> is a comma-separated list of ports or "all".

set-port-parameter|spp <port-list> priority|pri <value>
    Set the priority for the port(s) in <port-list>.
    <port-list> is a comma-separated list of ports or "all".
    Range for <value>: 0 - 127
```

### *3.4.2 Configuration*

From the **dec** subsystem prompt, the only necessary segment configuration step is to enable DECnet forwarding for all segments attached to DECnet networks. The **set-port-param mgmt-state enl** command tells the software that DECnet packets may arrive over these segments and that they should be used for routing purposes:

**set-port-parameter|spp** *<seg-list>* **mgmt-state|mst enl|dis**

This command can be used to either enable (**enl**) or disable (**dis**) DECnet forwarding for each segment. The command uses the normal <*seg-list*> syntax, which is a hyphen- and comma- separated list of segment numbers. For example, if segments 1, 2, and 3 are to be on DECnet networks, the command is:

**set-port-param 1-3 mgmt-state enl**

```
193:PowerHub:dec# spp 1-3 mst enl
Port 1: Okay
Port 2: Okay
Port 3: Okay
```

After enabling the segments, you can verify the segment configuration with the **display-port-param** command:

**display-port-param|dpp [***<seg-list>***|all]**

For example:

```
196:PowerHub:dec# dpp 1-2
DECnet port configuration (Port  1)
-----------------------------------
block-size:        1498
cost:              10
curr-adj-routers:  0
designated-rtr:    aa-00-04-00-1e-8a   (5.1023)
hello-time:        15 seconds
last-hello-sent:   12 seconds ago
mgmt-state:        Enabled
max-routers:       10
priority:          0
run-state:         Up
type:              Ethernet

DECnet port configuration (Port  2)
-----------------------------------
block-size:        1498
cost:              10
curr-adj-routers:  0
designated-rtr:    aa-00-04-00-1e-8a   (5.1023)
hello-time:        15 seconds
last-hello-sent:   12 seconds ago
mgmt-state:        Enabled
max-routers:       10
priority:          0
run-state:         Up
type:              Ethernet
```

At this point, you can also verify the routing status of the DECnet software through the **routing-status** command. This command shows the state of the global DEC forwarding system as well as whether or not each segment is configured to route DECnet packets.

```
198:PH-4:dec# rs
DECnet routing status:

Node/Port               Management State   Routing State
---------               ----------------   -------------
DEC-Forwarding          Enabled            Up

Port  1                 Enabled            Up
Port  2                 Enabled            Up
Port  3                 Enabled            Up
Port  4                 Disabled           Down
Port  5                 Disabled           Down
<remaining rows omitted for brevity>
```

In this listing, the `Management State` column refers to DECnet forwarding being enabled or disabled on each segment, while the `Routing State` column refers to the low-level hardware status. If that segment does not have a cable attached to it (and automatic segment-state detection is enabled), or if the segment has been disabled in the bridging subsystem, the `Routing State` shows "DOWN" instead of "UP."

## 3.5 DISPLAY COMMANDS

Using the display commands, you can:

- Look for adjacent DECnet routers in the network. (See Section 3.5.2 on page 78.)
- Look for all DECnet endnodes adjacent to the PowerHub system. (See Section 3.5.3 on page 78.)
- Look at the DECnet routing table to verify that all the routes are present. (See Section 3.5.4 on page 78.)

### 3.5.1 Verification of Routing

After the node and segments are configured, the PowerHub system begins forwarding packets among nearby nodes. To verify that the PowerHub system has identified its neighbors, you can use one of several display commands to examine routing tables and node lists. Figure 3–2 on page 77 shows a sample DECnet network. The display commands shown give information about this configuration. Note that we are monitoring the PowerHub system defined as node 5.1023, located on the left. It is serving as a Level-2 router for area 5, which consists of 7 nodes: itself, 5.34, 5.477, 5.45, 5.103, 5.553, and 5.811. There are 4 other areas, 37, 2, 8, and 59. In this picture, "endnodes" are depicted as a single circle. (Endnodes, such as non-routing workstations, are nodes not capable of forwarding packets.) Level-1 router nodes are shown as lightly-shaded rectangles.

Note that nodes that are capable of routing, but that appear on the periphery of networks (thus giving them nothing to route to), still qualify as routers and appear on the PowerHub listings as "routers" rather than "endnodes." Level-2 router nodes are rectangles, and are connected with bold lines. All connections to our PowerHub system are made through the segment numbers listed (1 through 5) by the small digits near the connecting lines. Also note that, while no endnodes are shown on the bold connections (links between departments, for example) between Level-2 routers, the protocols permit them to be there. For example, on the connection between 5.1023 and 37.322, endnodes or Level-1 routers for areas 5 and 37 could be attached. Each Level-2 router would recognize the nodes that belong to its area and forward packets to them.

.

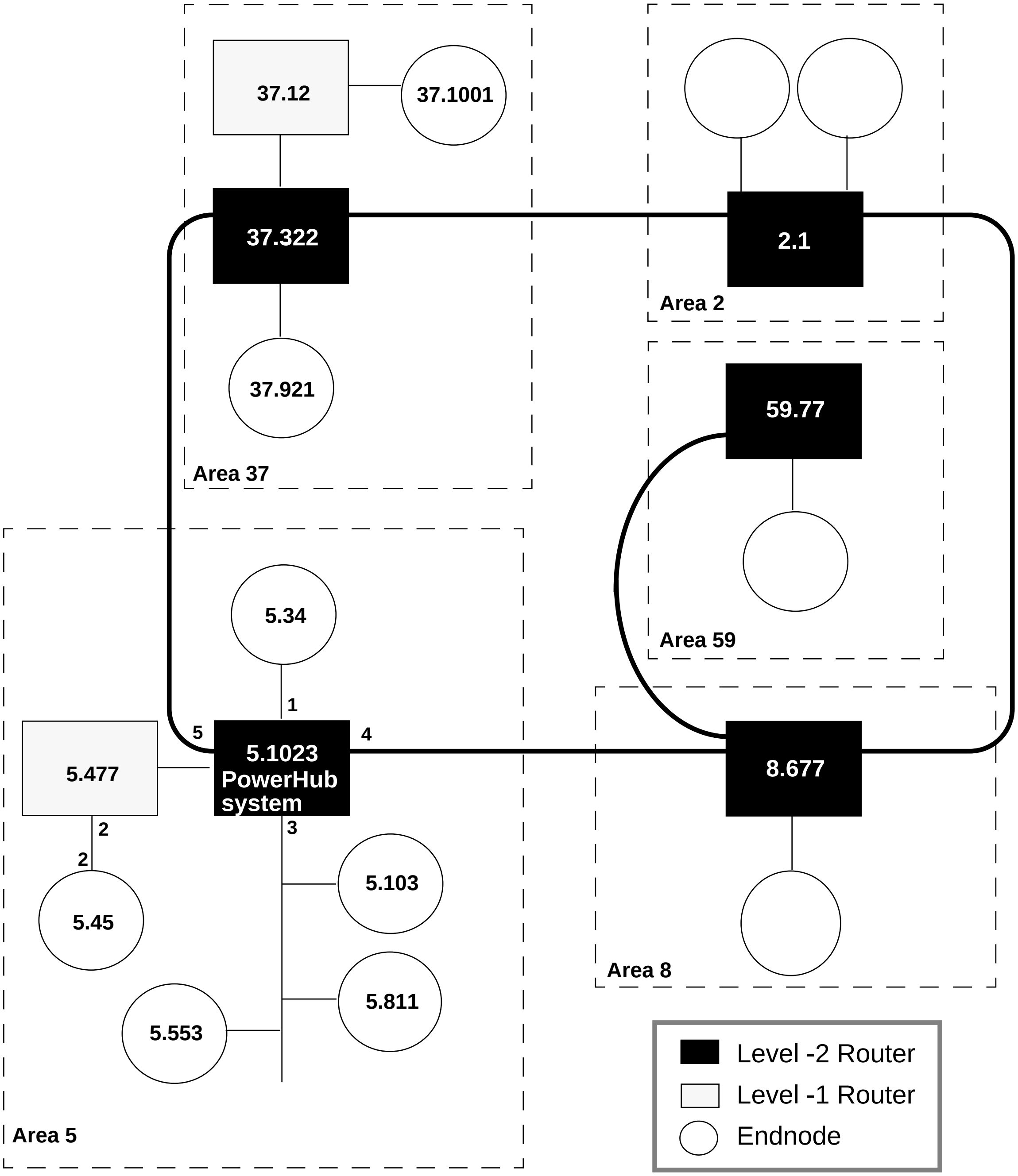


**FIGURE 3–2** Sample DECnet network.

### 3.5.2 Displaying Adjacent Routers

Look for adjacent routers in the network by typing:

**display-router-adj [***<node>***]**

```
407:PowerHub:dec# display-router-adj
DECnet router adjacency table:
Adj   Node ID  Type       State  Port  Blk Siz  Hello Tim  Priority  Age
----  -------  ---------  -----  ----  -------  ---------  --------  -----
1     5.477    Router     Up     2     1498     15         0         3
2     37.322   Area Rtr   Up     5     1498     15         0         3
3     8.677    Area Rtr   Up     4     1498     15         0         3
```

This command shows all the “adjacent” routers. In DECnet terminology, “adjacent” means “directly connected.” Thus nodes on the other end of 10Base-T links, or other sites on an Ethernet cable, are considered “adjacent.” A “router” is any node which can forward packets. Thus, this command shows all the directly connected routing nodes that the PowerHub system has discovered. One is inside the PowerHub system’s own area (5.477) and the other two are Level-2 routers (“`Area Rtr`”) in areas 37 and 8.

### 3.5.3 Displaying Adjacent Endnodes

Look for all endnodes adjacent to this router by typing:

**display-endnode-adj [***<node>***]**

```
408:PowerHub:dec# display-endnode-adj
DECnet end-node adjacency table:
Adj   Node ID  Type       State  Port  Blk Siz  Hello Tim  Priority  Age
----  -------  ---------  -----  ----  -------  ---------  --------  -----
1     5.34     End Node   Up     1     1498     10         0         9
2     5.811    End Node   Up     3     1498     10         0         9
3     5.103    End Node   Up     3     1498     10         0         9
4     5.553    End Node   Up     3     1498     10         0         9
```

This command shows all directly connected nodes that are “endnodes,” that is, those which cannot forward packets. The three nodes on the Ethernet cable are endnodes, as is node 5.34 (directly connected to segment 1). Note that node 5.45 is not adjacent, because this PowerHub system cannot reach it directly.

### 3.5.4 Displaying the Route Table

Look at the routing tables to verify that all the routes are present by issuing the **`display-route-tbl`** command. Here is the syntax for this command:

**display-route-tbl|drt [***<node>***|***<area-rtr>***]**

This command displays the route table, which is maintained by the DECnet routing software. It contains all the routes to nodes in this area that the PowerHub system has found dynamically. (DECnet does not provide for static, user-specified routes.)

Here is an example of the display produced by this command:

```
409:PowerHub:dec# display-route-tbl
DECnet routing table:
Node       Port   Next Hop           Hops  Cost
---------  -----  -----------------  ----  ----
area-rtr   -----  This-Rtr
5.34           1  ------                1    10
5.45           2  5.477                 2    10
5.103          3  ------                1    10
5.477          2  ------                1    10
5.553          3  ------                1    10
5.811          3  ------                1    10
5.1023     Local
```

Each entry contains the following information:

`Node` The address of the destination node.

`Port` The PowerHub segment that a packet destined for this node should leave on.

`Next Hop` The address of the next node a packet must pass through.

`Hops` The number of nodes the packet must pass through.

`Cost` A number reflecting the desirability of using this route.

From this table, we see that this area (Area 5) consists of 7 nodes: 34, 45, 103, 477, 553, 811, and 1023. The PowerHub system is node 1023. The nodes on the Ethernet cable are 103, 553, and 811; they are accessible directly through segment 3. Two other nodes, 34 and 477, can be contacted directly through segments 1 and 2. One node, number 45, can only be reached through node 477, which is a router. Therefore the routing table shows that to send packets to node 45, the “Next Hop” is node 477, and that the node is two hops away from this one.

Note that the PowerHub system, like all DECnet nodes, keeps track of the nearest Level-2 router. Since the PowerHub system is configured as an area-router (Level-2), the nearest Level-2 router is itself. Consequently, the next hop listed for the “`area-rtr`” node (the one responsible for all inter-area routing) says “`This-Rtr`.”

If this router is configured as an area router (Level-2), look at the area table. This is a list of all known areas, along with the best way to get to them. To display this table, issue the following command:

**`display-area-tbl [`*`<area>`*`]`**

```
410:PowerHub:dec# display-area-tbl
DECnet area table:
Area  Port   Next Hop           Hops  Cost
----  -----  -----------------  ----  ----
   2      4  8.677                 2    20
   5  Local
   8      4  8.677                 1    10
  37      5  37.322                1    10
  59      4  8.677                 2    20
```

From this table, we can tell that the PowerHub system is in area 5, and three other areas are accessible through the Level-2 router at 8.677, which is attached to the network on Segment 4. The other area is Area 37, available through segment 5.

If the PowerHub system is configured as a Level-1 router (in a multi-area network), the "`area-rtr`" entry points to another node. As an example, imagine that the other router (node 5.477) is also a PowerHub system.

If we were to examine the route table on that hypothetical node, you would see something like the following:

```
1:OtherPowerHub:dec# display-route-tbl
DECnet routing table:
Node       Port   Next Hop           Hops  Cost
---------  -----  -----------------  ----  ----
area-rtr       2  5.1023                1    10
5.34           2  5.1023                2    10
5.45           1  5.45                  1    10
5.103          2  5.1023                2    10
5.477      Local
5.553          2  5.1023                2    10
5.811          2  5.1023                2    10
5.1023         2  5.1023                1    10
```

Examine the node and segment statistics to verify that the PowerHub system is receiving data and control packets correctly. These commands are described in Section 3.5.5.

### *3.5.5 Other Display Commands*

Other display commands include the **`display-node-param`**, **`display-port-param`**, and **`display-routecache`** commands. The route cache holds the most recently used paths to various nodes. By examining the cache, you can determine which nodes were contacted most recently. The cache can be cleared with the **`flush-routecache`** command.

An example of how these commands are used is shown below:

```
100:PowerHub:dec# dc
DEC router cache:
Port 01: empty
Port 02: empty
Port 03: 30.331
101:PowerHub:dec# fc
Okay
102:PowerHub:dec# dc
DEC router cache:
Port 01: empty
Port 02: empty
Port 03: empty
```

## 3.6 DISPLAYING STATISTICS

There are two types of statistics collected in the DECnet subsystem: node statistics and segment statistics. The node statistics are displayed with the **`node-stats`** command, and they list information that is not associated with any particular segment. All the numbers displayed by the **`node-stats`** command relate to errors or dropped packets, so the ideal display is all zeros. Here is an example of the **`node-stats`** command:

```
411:PowerHub:dec# node-stats
DECnet node statistics (count since last stats clear):
node unreachable pkt loss    0
aged packet loss             0
node out-of-range pkt loss   0
oversized pkt loss           0
pkt format error             0
partial routing update loss  0
verification reject          0
routing table corrupted      0
no timers for updates        0
no bufs for sending hello    0
invalid hello from router    0
invalid hello from endnode   0
no room for router adj       0
no room for endnode adj      0
low priority rtr bumped      0
no bufs for lvl 1 update     0
lvl 1 msg format error       0
lvl 1 msg checksum error     0
lvl 1 msg area num error     0
no bufs for lvl 2 update     0
lvl 2 msg format error       0
lvl 2 msg checksum error     0
router moved to diff. port   0
end node moved to diff. port 0
```

As in other PowerHub subsystems, the DECnet software maintains two copies of the node statistics:

- Count since the last clear.
- Count since the last system reset.

Both counters increment when errors occur, but the **node-stats-clear** command clears only the count since last clear. To display the count since the last reset, use the **-t** option with the **node-stats** command, as shown in the following example. In this particular example, the hub has just been rebooted and no statistics have yet been collected.

```
412:PowerHub:dec# node-stats -t

DECnet node statistics (count since last stats reset):
node unreachable pkt loss    0
aged packet loss             0
node out-of-range pkt loss   0
oversized pkt loss           0
pkt format error             0
partial routing update loss  0
verification reject          0
routing table corrupted      0
no timers for updates        0
no bufs for sending hello    0
invalid hello from router    0
invalid hello from endnode   0
no room for router adj       0
no room for endnode adj      0
low priority rtr bumped      0
no bufs for lvl 1 update     0
lvl 1 msg format error       0
lvl 1 msg checksum error     0
lvl 1 msg area num error     0
no bufs for lvl 2 update     0
lvl 2 msg format error       0
lvl 2 msg checksum error     0
router moved to diff. port   0
endnode moved to diff. port  0
```

The segment statistics are collected in the same manner. These statistics are primarily counts of how many DECnet packets are routed through each segment. This can give you an idea of where the most traffic is coming from, and may provide insight on how to better structure the network.

The **port-stats** command displays the count since last clear. To display the count since last reset, use the **-t** option. This command uses the standard *<seg-list>* format, which consists of a comma- and hyphen-separated list of segment numbers (ex: **1,4,5-7**). The **port-stats** command provides the following kinds of information.

```
413:PowerHub:dec# port-stats 1,2

DECnet packet statistics for port 1 (count since last stats clear):
transit packet received      0
terminating packet received  61716
hello pkts rcvd              43007
Level 1 update msgs rcvd     10944
Level 2 update msgs rcvd     7765
pkts rcvd from self          0
transit packet sent          0
originating packet sent      63968
transit congestion loss      0
terminating congestion loss  0
circuit down                 0
pkts not sent                0
end node adjacency down      0
router adjacency down        0
end node not added to rt tbl 0
end node is not in rt tbl    0

DECnet packet statistics for port 2 (count since last stats clear):
transit packet received      0
terminating packet received  34402
hello pkts rcvd              34402
Level 1 update msgs rcvd     0
Level 2 update msgs rcvd     0
pkts rcvd from self          0
transit packet sent          0
originating packet sent      64034
transit congestion loss      0
terminating congestion loss  0
circuit down                 0
pkts not sent                0
end node adjacency down      0
router adjacency down        0
end node not added to rt tbl 0
end node is not in rt tbl    0
```

To clear the count since last clear, issue the **`port-stats-clear`** command. This command can take the *`<seg-list>`* argument, to display certain segments, or the keyword **`all`**, to clear statistics for all segments.

# 4 IP Security Commands

This chapter describes the PowerHub software implementation of the IP security options specified by RFC 1108, “U.S. Department of Defense Security Options for the Internet Protocol.”

RFC 1108 describes how the Internet protocol implements the U.S. Department of Defense Basic Security Option (BSO) and the Extended Security Option (ESO). For more complete information, consult this RFC.

Both options (BSO and ESO) are encoded in the *options* area of each IP datagram. These options are used to:

- Standardize the representation and transmission of labels required by network security models.
- Validate a datagram as appropriate for transmission from the source and delivery to the destination.
- Ensure that the datagram travels over an appropriately protected route (provided that routing protocols implement security label information).

## 4.1 BASIC SECURITY OPTION

The *Basic Security Option (BSO)* carries classification level and protection authority flags. Figure 4–1 shows the BSO format.

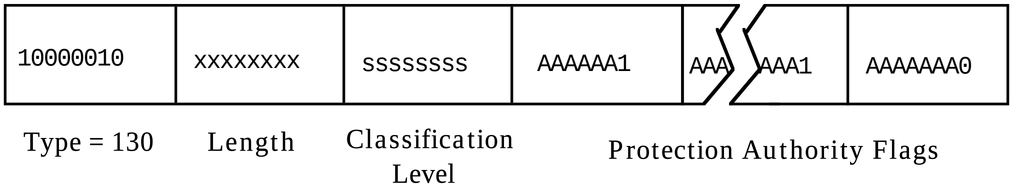


**FIGURE 4–1** Format of Basic Security Option.

As shown in Figure 4–1, this option has the following fields:

`Type`
The binary value `10000010` (decimal `130`) identifies the Basic Security Option.

`Length`
This bit field specifies the length of the option in octets. The length is variable and must be at least three octets, including the `Type`, `Length`, and `Classification Level` fields, which must be present. The `Protection Authority Flags` field is optional.

Classification Level
This one-octet field specifies the datagram's classification level. The encodings are shown in Table 4–1 from highest classification level to lowest. Note that the encodings have been chosen so that any two classification levels differ in at least four bit positions. Values labeled "reserved" are considered invalid until they have been assigned to named classification levels.

**TABLE 4–1** Classification-level encoding.

| Encoding | Classification Level |
|---|---|
| 00000001 | Reserved 4 |
| 00111101 | Top Secret |
| 01011010 | Secret |
| 10010110 | Confidential |
| 01100110 | Reserved 3 |
| 11001100 | Reserved 2 |
| 10101011 | Unclassified |
| 11110001 | Reserved 1 |

Protection Authority Flags

This field specifies rules for transmission and processing of the information in the datagram. The field consists of a variable number of octets, with the most significant bit in each octet coming first (“big-endian”[1] format). Figure 4–2 shows the big-endian bit order.

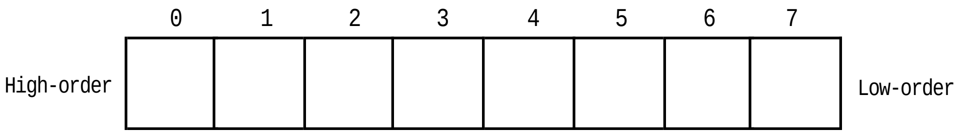


**FIGURE 4–2** “Big-Endian” Bit Order.

The first seven bits (bits 0 through 6) of each octet are flags. Each flag is associated with a protection authority.[2] If the bit is `1`, then the datagram must be protected in accordance with the rules of that authority.

Table 4–2 shows the protection authority flags currently assigned. The low-order bit (bit 7) of each octet is encoded as `0` if it is the final octet in the field, or `1` if there are additional octets. Trailing all-zero octets are prohibited.

Initially this field requires only one octet, because there are fewer than seven protection authorities currently defined.

**TABLE 4–2** Protection authority bit assignments.

| Bit Number | Authority |
|---|---|
| 0 | GENSER |
| 1 | SIOP-ESI |
| 2 | SCI |
| 3 | NSA |
| 4 | DOE |
| 5,6 | unassigned |
| 7 | field termination indicator |

[1] The terms “little-endian” and “big-endian” originate in Jonathan Swift’s *Gulliver’s Travels*, where they identify two violently antagonistic religious factions that break their eggs at the little end and the big end respectively.

[2] RFC 1108 defines security labels specifying the type of protection a datagram must receive. Actually providing this protection lies outside the scope of the IP protocols.

## 4.2 BASIC-SECURITY-OPTION CONFIGURATION PARAMETERS

When you configure the PowerHub system for IP Security, you can define system-level (global) parameters and segment-specific parameters. Each parameter has a default value, which is reinstated under any of the following conditions:

- Power is cycled off and on.
- The **`security user-interface`** command, described in Section 4.5.2, is issued.

One parameter, **`control`**, is general to IP Security and is not associated exclusively with system-level or segment-level security. This parameter specifies whether IP Security is activated. The default is “disabled.” The system-level and segment-level parameters are described in the following sections.

### 4.2.1 System-Level (Global) Parameters

Table 4–3 lists the configuration parameters that must be defined for each PowerHub system implementing the BSO. The parameter names match the names you specify with the **`security`** command when you define them. For each parameter, the default value is listed. See the procedures in Section 4.8 for information on defining these parameters

**TABLE 4–3** System-level parameters.

| Parameter and Description | Default |
|---|---|
| **`system-authority-in\|sai`**<br>The set of all Protection-Authority-Flag fields permitted in any datagram received by the PowerHub system. Each set element is a series of one or more octets, representing a permitted combination of protection-authority flags.<br>For instance, if **`sai`** contains the elements `00110000` and `01011000`, then received datagrams protected according to SCI and NSA rules, in the first case, or SIOP-ESI, NSA, and DOE rules, in the second, are permitted. | none |
| **`system-authority-out\|sao`**<br>The set of all Protection-Authority-Flag fields permitted in any datagram transmitted by the PowerHub system. Each set element is a series of one or more octets, representing a permitted combination of protection-authority flags. | none |
| **`system-level-max\|slx`**<br>The highest classification level permitted in any datagram transmitted by the PowerHub system. | unclassified |
| **`system-level-min\|sln`**<br>The lowest classification level permitted in any datagram transmitted by the PowerHub system. | unclassified |

### 4.2.2 Segment-Level Parameters

Table 4–4 lists the configuration parameters that can be defined for each PowerHub segment. These parameters further restrict, for the segment(s) you specify, the corresponding system-level parameters.

As with the system-level parameters, the names match the names you specify with the security command when you define them. For each parameter, the default value is listed. See the procedures in Section 8.9 for information on defining these parameters.

**TABLE 4–4** Segment-level parameters.

| Parameter and Description | Default |
|---|---|
| **`port-authority-err|pae`**<br>A single Protection-Authority-Flag field, selected from the values in **`port-authority-out`** and assigned to transmitted ICMP error messages. | none |
| **`port-authority-in|pai`**<br>The set of all protection authority flag fields permitted in any datagram received on this segment. | none |
| **`port-authority-out|pao`**<br>The set of all protection authority flag fields permitted in any datagram transmitted on this segment. | none |
| **`port-bso-reqd-recv|pbr`**<br>A `0` (false or no) or `1` (true or yes) indicating whether all datagrams received on this segment must contain a Basic Security Option. If `0` (no), unlabeled datagrams receive the classification label and protection authority flag field specified in **`port-implicit-label`**; if `1` (yes), receipt of an unlabeled datagram results in an error. | **`no`** |
| **`port-bso-reqd-xmit|pbx`**<br>A `0` (false or no) or `1` (true or yes) indicating whether all datagrams transmitted on this segment must contain a Basic Security Option. If `0` (no), **`port-bso-reqd-xmit`** should also be set to **`no`**. | **`no`** |
| **`port-def-auth-out|pdao`**<br>Sets or clears a single Default-Protection-Authorities field attached to unlabeled datagrams transmitted from the specified segment. This is not an RFC 1108 parameter but gives you the ability to configure default security settings for outgoing packets. | none |
| **`port-implicit-label|pil`**<br>A single classification level and a single protection authority flag field to be associated with all "unlabeled" datagrams received on this segment. An *unlabeled* datagram is one that does not contain a Basic Security Option. The **`port-implicit-label`** setting is meaningful only if **`port-bso-reqd-recv`** is false. | level: unclassified<br>authorities: none |

**TABLE 4–4** (Continued) Segment-level parameters.

| **Parameter and Description** | **Default** |
|---|---|
| **`port-level-max\|plx`**<br>The highest classification level permitted in any datagram transmitted or received on this segment. | unclassified |
| **`port-level-min\|pln`**<br>The lowest classification level permitted in any datagram transmitted or received on this segment. | unclassified |
| **`port-strip-bso\|psb`**<br>Enables you to remove the Basic Security Option from packets going out on the specified segment or segments. This feature is particularly useful if you are sending to a system that does not implement security labeling. | **`no`** |

### 4.2.3 Required Security Relationships

In every PowerHub system implementing IP Security, the following relationships must hold:

$$\mathtt{system}-\mathtt{level}-\mathtt{max} \geq \mathtt{port}-\mathtt{level}-\mathtt{max} \geq \mathtt{port}-\mathtt{level}-\mathtt{min} \geq \mathtt{system}-\mathtt{level}-\mathtt{min}$$

$$\mathtt{system}-\mathtt{authority}-\mathtt{in} \supseteq \mathtt{port}-\mathtt{authority}-\mathtt{in}$$

$$\mathtt{system}-\mathtt{authority}-\mathtt{out} \supseteq \mathtt{port}-\mathtt{authority}-\mathtt{out}$$

For IP Security to function properly on the PowerHub system, your IP Security definitions must adhere to these relationships.

## 4.3 EXTENDED SECURITY OPTION

The Extended Security Option (ESO) has not yet been fully defined and is therefore not currently implemented by PowerHub software. This option will permit a datagram to carry additional security information beyond that in the Basic Security Option. RFC 1108 specifies the ESO format. Additional security information codes, along with the associated syntax, semantics, processing rules, and configuration parameters, are to be published in future RFCs.

## 4.4 IP SECURITY COMMANDS

All IP security commands are part of the **ip** subsystem. To issue these commands you must be in the **ip** subsystem, or you must preface the command with "`ip`". All security commands begin with **`security`** (or its terse form **`sec`**).

The PowerHub system lets you choose between two different levels of security user interface: extended or simple. The *extended user interface* lets you individually set all the IP security parameters listed in Section 4.2 on page 88. The *simple user interface* lets you define the following security parameters on a segment-by-segment basis:

- **`port-level-max`**
- **`port-level-min`**
- **`port-authority-in`** and **`port-authority-out`** (must use the same flag list for both incoming and outgoing datagrams).

The display that follows shows the on-line help display for the extended user interface.

```
security|sec <parameter-and-info>
    Command to configure system for handling IP security options.
    There are two forms of <parameter-and-info>:
          <security-parameter> <security-info>
          <port-list> <security-parameter> <security-info>
    <security-info> and the presence/absence of <port-list> vary
    based on <security-parameter>.  For more information, use help
    specific to <security-parameter>, one of the following:
        control | ctl                    license | lic
        port-configuration | pcf         secure | sec
        system-configuration | scf       unsecure | unsec
        user-interface|ui
```

As shown in these help listings, the first argument of this command (for system-wide parameters) or the second argument (for segment-level parameters) indicates the parameter to be configured. The remaining arguments specify the desired configuration.

You can change between user interfaces using the **`security user-interface`** command. Note, however, that ***all*** security parameters are reset to their default values when you use this command.

## 4.5 ENABLING IP SECURITY

Enabling the PowerHub system's IP security processing involves these steps:

(1) Enter a valid IP Security license number. (The license number is supplied when you purchase your IP Security license from FORE Systems.)

(2) Enable IP security control.

(3) Optionally change the user-interface level.

### *4.5.1 Entering a License Number*

To enter your IP Security license:

(1) Access the **ip** subsystem, if you have not already done so. (To access the **ip** subsystem, issue the **`ip`** command.)

(2) Enter the following command at the command prompt:

**`sec license`** *`<num1>-<num2>-<num3>`*

where:

*`<num1>`* Specifies the option for which you are entering a license number. Specify **01** for IP security.

*`<num2>`* Specifies the IP security serial number.

*`<num3>`* Is a unique 10-digit number used with the serial number to identify your IP Security license.

Here is an example of this command:

```
6: PowerHub:ip#  security license 01-1234-0123456789
7: PowerHub:ip#
```

In the command shown in the example above, you must replace the sample license number `01-1234-0123456789` with your own factory-supplied license number.

The **`license`** command activates the IP security features and initializes IP security parameters (other than **`license`**) to their default values, but does not enable IP Security. (This is accomplished using the **`security control`** command, discussed in Section 4.2 on page 88.)

Another form of the **`license`** command, **`security license clear`**, invalidates an existing license. This command is accepted only when a valid license has already been entered.

### *4.5.2 Enabling IP Security Control*

You can enable IP security processing of incoming and outgoing packets using the **`sec control`** command, as shown in the following example:

```
7:PowerHub:ip# security control enl
8:PowerHub:ip#
```

When security processing is enabled by this command, all incoming packets are examined for security options and are blocked or forwarded according to the default security parameters or others you have configured. All outgoing packets have security options added, deleted, or modified according to the appropriate parameters. The default value is **`dis`** (disabled).

The **`security control`** command can also disable IP security processing.

```
8:PowerHub:ip# security control dis
9:PowerHub:ip#
```

When IP security processing is disabled, incoming security options are ignored, and security options for outgoing packets are not added, deleted, or modified.

Note that disabling IP security processing does not affect the current settings of any IP security parameters. If you make changes to IP security parameters while security processing is disabled, the changes take effect the next time security processing is enabled. We recommend that you configure all security parameters before enabling security processing.

### 4.5.3 Changing the User Interface Level

Before configuring the PowerHub system for IP Security, you can choose between the following user-interface levels:

*Simple* Contains few parameters. The IP Security parameters are arguments specified with the **`sec secure`** command. These parameters apply only to individual segments. If you want to implement system-wide security, we recommend that you use the extended interface.

*Extended* Lets you set each IP Security parameter on a system-wide basis.

To change the user interface level, issue the following command:

**`sec user-interface|ui simple|sim | extended|ext`**

where:

**`simple|sim | extended|ext`**

Specifies whether you want the simple user interface or the extended one. The default is **`simple`**.

> **NOTE**: When you issue the **`user-interface`** command, all IP Security parameters are reset to their defaults. We recommend that you issue this command from within your configuration file. Place the command before the commands that set the IP Security parameters.

## 4.6 SAVING THE IP SECURITY CONFIGURATION

To activate IP security, the **security control** command must be issued every time the PowerHub system reboots. In addition, the IP security parameters described in later subsections must be set to their desired values. You can issue these commands manually each time you reboot the PowerHub system, or you can save them to a configuration file.

To save the IP Security configuration settings to a configuration file, issue the **mgmt svcfg** command or the **tftp svcfg** command, followed by the configuration file name. Here is an example:

```
9:PowerHub:ip# mgmt svcfg cfg
Configuration saved to cfg
10:PowerHub:ip#
```

In this example, the default configuration file name (`cfg`) is specified. The following information is saved in the configuration file:

- The license number.
- The current state of IP security processing (enabled or disabled by the **security control** command).
- The current value of all configurable IP security parameters.
- The user interface setting (simple or extended).

If the configuration is stored in the file named `cfg`, it is restored each time the system is rebooted. Like other configuration information, this information is stored in ASCII format and can be edited off-line. See the *Installation and Configuration Manual* for your PowerHub system.

## 4.7 USING THE SIMPLE INTERFACE

If you choose to use the simple interface, use the following command to configure specific segments for IP Security:

**security|sec** *<seg-list>* **secure|sec** *<min-level>*

**[to** *<max-level>***]** **[***<auth-list>* **[***<auth-list>...***]]**

where:

*<seg-list>* Specifies the segments for which you are setting the security parameters.

*<min-level>*

Specifies the minimum acceptable security level of incoming or outgoing datagrams. You can specify one of the following:

**top-secret|ts**

**secret|se**

**confidential|cf**

**unclassified|uc**

The default is **unclassified**.

**to** *<max-level>*

Specifies the maximum acceptable security level of incoming or outgoing datagrams. You can specify one of the following, provided it is greater than or equal to the value specified for <min-level>:

**top-secret|ts**

**secret|se**

**confidential|cf**

**unclassified|uc**

The default is the value specified for *<min-level>*.

*<auth-list>*

Specifies the specific fields you are adding or removing. Valid fields are:

**genser|gs**

**siop-esi|se**

**sci|sc**

**nsa|ns**

**doe|do**

Use commas to separate the flags within each field of the *<flag-list>* argument. Use spaces to separate different fields. Do not include a space before or after a comma, or the PowerHub system will interpret it as a field separator.

To remove the security parameters from one or more segments, use the following command:

```
security <seg-list> unsecure|unsec
```

## 4.8 SYSTEM-LEVEL IP SECURITY PARAMETERS AND COMMANDS

> **NOTE**: The security parameters described in this section appear only in the extended user interface. To access this user interface, use the **`user-interface`** command. See Section 4.5.3 on page 93.

The PowerHub system's IP security software lets you set security parameters at both the system-wide (PowerHub) level and at the segment level, corresponding to the requirements of RFC 1108. In general, the system-wide parameters specify the highest (most secure) and lowest (least secure) levels of authority permitted. Segment-level parameters must be set at or within the system-level bounds.

This subsection describes the commands for setting system-wide parameters. These commands do not take a *`<seg-list>`* argument. The remainder of this section describes the system-wide IP security parameters and corresponding commands in detail. The commands and examples in this section assume that you are using the extended user interface.

The display that follows shows the on-line help for the system-level IP Security commands.

```
security|sec control|ctl dis|enl
    Disable/Enable IP security option processing.

security|sec system-authority-in|sai add|del <authlst>
    Add/Delete protection authority flag fields used for validating
    incoming datagrams.

    <authlst> is a comma-separated list of authority flags:
       genser | gs      siop-esi | se      sci | sc
       nsa | ns        doe | do

security|sec system-authority-in|sai none
    Remove all protection authority flag fields used for validating
    incoming datagrams.

    WARNING: This will remove authority flag fields in port-authority-in
    for all ports.

security|sec system-authority-out|sao add|del <authlst>
    Add/Delete protection authority flag fields used for validating
    outgoing datagrams.

    <authlst> is a comma-separated list of authority flags:
        genser | gs      siop-esi | se      sci | sc
        nsa | ns        doe | do
```

```
security|sec system-authority-out|sao none
    Remove protection authority flag fields used for validating outgoing
    datagrams.

    WARNING: This will remove authority flag fields in port-authority-out
    for all ports.

security|sec system-level-max|slx <level>
    Specify highest classification level that may be present in either
    incoming or outgoing datagrams.

    <level> is one of the following:
        top-secret | ts        secret | se
        confidential | cf     unclassified | uc

security|sec system-level-min|sln <level>
    Specify lowest classification level that may be present in either
    incoming or outgoing datagrams.

    <level> is one of the following:
        top-secret | ts        secret | se
        confidential | cf      unclassified | uc
```

### 4.8.1 Displaying the System Security Configuration

At any time, you can display the IP Security configuration for the PowerHub system using the **sec system-configuration** command. Here is an example of the display produced by this command:

```
10:PowerHub:ip# sec scf
................ IP Security System Configuration .................

Control:              Disabled
system-authority-in:  sci nsa,doe
system-authority-out: none
system-level-max:     unclassified
system-level-min:     unclassified
11:PowerHub:ip#
```

To display the IP Security configuration for a specific segment or segment list, use the **sec port-configuration** command, described in Section 4.10.1 on page 103.

### 4.8.2 Defining Protection-Authority-Flag Fields

The Protection-Authority-Flag fields need to be defined for incoming and outgoing datagrams. Use the procedures in the following sections to add or remove Protection-Authority-Flag fields.

One of the fields must be present in its entirety in each incoming packet on every segment. If no valid field is present, the packet is not forwarded.

You can specify more restrictive sets of flags for individual segments using the **security port-authority-in** command, described in Section 4.10.3.1 on page 104.

#### *4.8.2.1 Incoming Datagrams*

To add or delete Protection-Authority-Flag fields for incoming datagrams, issue the following command:

```
security|sec system-authority-in|sai
                 add|del <auth-list>
```

where:

**add|del** Specifies whether you are adding or removing the specified fields.

*<auth-list>* Specifies the specific fields you are adding or removing. Valid fields are:

**genser|gs**

**siop-esi|se**

**sci|sc**

**nsa|ns**

**doe|do**

Use commas to separate the flags within each field of the *<auth-list>* argument. Use spaces to separate different fields. Do not include a space before or after a comma, or the PowerHub system will interpret it as a field separator.

For example, to specify that all incoming packets must contain either the **sci** flag or both the **doe** and the **nsa** security flags, type the following command:

```
11:PowerHub:ip# sec sai add sci doe,nsa
12:PowerHub:ip#
```

To verify flags, use the security system-configuration command:

```
12:PowerHub:ip# sec scf
................. IP Security System Configuration .................

Control:              Disabled
system-authority-in:  sci nsa,doe
system-authority-out: none
system-level-max:     unclassified
system-level-min:     unclassified
13:PowerHub:ip#
```

You can enter flags in any order, but the **scf** command always lists flags within a field in the order **genser**, **siop-esi**, **sci**, **nsa**, **doe**, corresponding to the RFC 1108 bit assignments.

If you specify both valid and invalid flag fields, only the valid fields are added.

To specify that all incoming packets must contain both **genser** and **sci** flags, or else **nsa**, **doe**, and **siop-esi** flags, issue the following command:

```
13:PowerHub:ip# sec sai add gs,sc ns,do,se
14:PowerHub:ip#
```

Note that the **sec sai add** command adds to the existing list of authority flag fields. It does not delete or replace flags or fields already on the list. Thus, the list resulting from the execution of commands 11 and 13 above has five fields.

```
14:PowerHub:ip# sec scf
................ IP Security System Configuration ................
Control:              Disabled
system-authority-in:  genser,sci siop-esi,nsa,doe sci nsa,doe
system-authority-out: none
system-level-max:     unclassified
system-level-min:     unclassified
15:PowerHub:ip#
```

To delete one or more flag fields from the current list, use the **sec sai del** command. The flag fields to be deleted, which you specify as an argument, must be present in the current configuration. You can enter the flags within a field in any order. If you specify both valid and invalid fields, only the valid fields are deleted. You cannot add or delete flags within a field.

To delete all current flag fields, issue the following command:

**sec system-authority-in none**

This command removes all Protection-Authority-Flag fields used to validate incoming datagrams. Deleting the system-wide fields also deletes all segment-level flag fields.

If an incoming packet has no security option label, its treatment depends on the segment on which it arrives. If the **port-bso-reqd-recv** parameter is set to **yes**, the packet is not forwarded. However, if this parameter is set to **no**, the packet is assigned the security options specified for the **port-implicit-label** parameter and processed accordingly.

#### *4.8.2.2 Outgoing Datagrams*

To add or delete Protection-Authority-Flag fields for outgoing datagrams, issue the following command:

```
security|sec system-authority-out|sao
             add|del <auth-list>
```

where:

**add|del** Specifies whether you are adding or removing the listed fields.

*<auth-list>* Specifies the fields you are adding or removing. Valid fields are:

**genser|gs**

**siop-esi|se**

**sci|sc**

**nsa|ns**

**doe|do**

As with the **security system-authority-in** command, you must use commas to separate the flags within each field of the *<flag-list>* argument. Use spaces to separate different fields. Do not include a space before or after a comma, or the PowerHub system will interpret it as a field separator.

The **security system-authority-out** command specifies a set of Protection-Authority-Flag fields. Each field is a combination of protection authority flags. One of these fields must be present in its entirety in each outgoing (forwarded) packet on every segment. If no valid field is present, the packet is not transmitted. You can specify more restrictive sets of flags for individual segments using the **security port-authority-out** command, described in Section 4.8.3.2 on page 101. The default is **none** (no protection authority flags may be present).

If an outgoing (forwarded) packet has no security option label, its treatment depends on the segment on which it is transmitted. If the **port-bso-reqd-xmit** parameter is set to **yes**, the packet is not forwarded. However, if this parameter is set to **no**, the unlabeled packet is forwarded without modifications (except that any security fields present are stripped).

To remove all Protection-Authority-Flag fields for outgoing datagrams, issue the following command:

```
sec system-authority-out none
```

Deleting the system-wide fields also deletes all segment-level flag fields.

### *4.8.3 Specifying the Classification Levels*

Use the commands in this section to specify the maximum and minimum classification levels permitted in any datagram transmitted or received by the PowerHub system. You can specify more restrictive levels for individual segments using the **sec port-level-max** and **sec port-level-min** commands described in the next subsection.

#### *4.8.3.1 Maximum Classification Level*

To specify the highest classification level permitted in any datagram transmitted or received by the system, use the following command:

**`security|sec system-level-max|slx`** *`<level>`*

where:

| | |
|---|---|
| *`<level>`* | Is one of the following:<br>**`top-secret|ts`**<br>**`secret|se`**<br>**`confidential|cf`**<br>**`unclassified|uc`**<br>The default is **`unclassified`**. |

#### *4.8.3.2 Minimum Classification Level*

To specify the lowest classification level permitted in any datagram transmitted or received by the system, use the following command:

**`security|sec system-level-min|sln`** *`<level>`*

The values for the *`<level>`* arguments are identical to those for the **`system-level-max`** command. The *`<level>`* you specify for the **`system-level-min`** parameter must be less than or equal to the **`system-level-max`** parameter.

## *4.9 SEGMENT-LEVEL COMMANDS*

> **NOTE**: The security parameters described in this section appear in the extended user interface. If you want to use the simple user interface, see Section 4.5.3 on page 93.

Help listings for security commands that apply to a *`<seg-list>`* are listed in Table 4–5. As in other commands, *`<seg-list>`* is a comma- and hyphen-separated list of segments, or the keyword **`all`**. Note that *`<seg-list>`* always follows the **`sec`** command and precedes the segment-level command.

## 4.10 SEGMENT-SPECIFIC IP SECURITY COMMANDS

Table 4–5 lists the segment-specific IP Security commands.

**TABLE 4–5** Segment-specific IP security commands.

```
security|sec <port-list> port-authority-err|pae none|<authlst>
    Set/Clear authorities flag used in outgoing ICMP Error/Response Messages
    for specified port(s).
    NOTE: This parameter must a subset of port-authorities-out.
    <authlst> is a comma-separated list of authority flags:
        genser | gs      siop-esi | se      sci | sc
        nsa | ns         doe | do

security|sec <port-list> port-authority-in|pai none
    Remove all protection authority flag fields used for validating incoming
    datagrams on specified port(s).

security|sec <port-list> port-authority-in|pai add|del <authlst> [<authlst> ...]
    Add/Delete protection authority flag fields used for validating incoming
    datagrams on specified port(s).
    NOTE: port-authority-in must be a subset of system-authority-in.
    <authlst> is a comma-separated list of authority flags:
        genser | gs      siop-esi | se      sci | sc
        nsa | ns         doe | do

security|sec <port-list> port-authority-out|pao none
    Remove all protection authority flag fields used for validating outgoing
    datagrams on specified port(s).

security|sec <port-list> port-authority-out|pao add|del <authlst>
    Add/Delete protection authority flag fields used for validating outgoing
    datagrams on specified port(s).
    NOTE: port-authority-out must be a subset of system-authority-out.
    <authlst> is a comma-separated list of authority flags:
        genser | gs      siop-esi | se      sci | sc
        nsa | ns         doe | do

security|sec <port-list> port-bso-reqd-recv|pbr yes|y|no|n
    Specify whether basic security option must be present in incoming
    datagrams on specified port(s).

security|sec <port-list> port-bso-reqd-xmit|pbx yes|y|no|n
    Specify whether basic security option must be present in outgoing
    datagrams on specified port(s).
security|sec <port-list> port-configuration|pcf
    Display security parameters for specified port(s).

security|sec <port-list> port-def-authority-out|pdao none|<authlst>
    Set/Clear default authorities flag used in outgoing datagrams
    for specified port(s).
    NOTE: This parameter must a subset of port-authorities-out
    <authlst> is a comma-separated list of authority flags:
        genser | gs      siop-esi | se      sci | sc
        nsa | ns         doe | do
```

**TABLE 4–5** (Continued) Segment-specific IP security commands.

```
security|sec <port-list> port-implicit-label|pil <level> [none|<authlst>]
    Security label used for validating incoming datagrams on specified
    port(s), that do not have the basic security option.
    NOTE: This parameter is applicable only if port-bso-reqd-recv is
    "no" for the specified port(s).
    <level> is one of the following:
        top-secret | ts       secret | se
        confidential | cf     unclassified | uc
    <authlst> is a comma-separated list of authority flags:
        genser | gs      siop-esi | se      sci | sc
        nsa | ns         doe | do

security|sec <port-list> port-level-max|plx <level>
    Specify highest classification level that may be present in either
    incoming or outgoing datagrams on specified port(s).
    <level> is one of the following:
        top-secret | ts       secret | se
        confidential | cf     unclassified | uc

security|sec <port-list> port-level-min|pln <level>
    Specify lowest classification level that may be present in either
    incoming or outgoing datagrams on specified port(s).
    <level> is one of the following:
        top-secret | ts       secret | se
        confidential | cf     unclassified | uc

security|sec <port-list> port-strip-bso|psb no|n|yes|y
    Strip BSO in packets going out on the specified port(s).
    NOTE: This parameter is applicable only if port-bso-reqd-xm
 is set to "no".
```

### *4.10.1 Displaying the Segment-Security Configuration*

At any time, you can display the IP Security configuration for one or more specific segments using the **sec port-configuration** command. Here is the syntax for this command:

**security|sec** *<seg-list>* **port-configuration|pcf**

Here is an example of the display produced by this command. In this example, segment 1 is specified for *<seg-list>*.

```
14:PowerHub:ip# sec 1 pcf

.......... IP Security System Configuration for Port 1...........

port-authority-err:  none
port-authority-in:  none
port-authority-out:  none
port-bso-reqd-rec:  no
port-bso-reqd-xmit:  no
port-def-auth-out:  none
port-implicit-label:  Lvl:  unclassified    Auth:  none
port-level-max:  unclassified
port-level-min:  unclassified
port-strip-bso:  yes

15:PowerHub:ip#
```

### 4.10.2 Specifying a Protection Authority Flag Field

The **`port-authority-err`** parameter specifies a single Protection-Authority-Flag field. This field is attached to ICMP error messages transmitted from the specified segment or segments. This error field must be one of those present in the **`port-authority-out`** parameter for the segment, which in turn must be present in the **`system-authority-out`** parameter. The default is **`none`** (no protection authorities specified).

The syntax for this command is:

```
security|sec <seg-list>
                port-authority-err|pae
                none|<auth-list>
```

where:

*`<seg-list>`* Specifies the segment(s) for which you are defining the Protection-Authority-Flag field.

**`none|`***`<auth-list>`* Specifies the contents of the Protection-Authority-Flag field. You can specify **none** or a list containing any of the following:

**`genser|gs`**

**`siop-esi|se`**

**`sci|sc`**

**`nsa|ns`**

**`doe|do`**

You can specify only a single *`<flag-list>`* field. To set **`port-authority-err`** to a different value, repeat the command with the new value. To clear the current value, use **`none`** as the argument.

### 4.10.3 Defining a Set of Protection-Authority-Flag Fields

You can define a set of Protection-Authority-Flag fields for incoming and outgoing datagrams. Incoming and outgoing fields are defined using different commands. The following sections describe these commands.

#### 4.10.3.1 Incoming Datagrams

The **`security`** **`port-authority-in`** command adds to, deletes from, or clears a set of Protection-Authority-Flag fields. Each field *`<auth-list>`* is a list of Protection-Authority flags. One of these fields must be present in its entirety in each incoming packet on the specified segment or segments. If no valid field is present, the packet is not forwarded.

The flags specified for *`<auth-list>`* must be a subset of the flags specified for *`<auth-list>`* with the **`system-authority-in`** command. The default is **`none`** (no protection authority flags may be present).

The syntax for the **`port-authority-in`** command is identical to that for **`system-authority-in`**, except for the addition of a *`<seg-list>`*.

#### *4.10.3.2 Outgoing Datagrams*

The **sec port-authority-out** command adds to, deletes from, or clears a set of Protection-Authority-Flag fields. One of these fields must be present in its entirety in each outgoing (forwarded) packet on the specified segment or segments. If no valid field is present, the packet is not transmitted.

The flags specified for *<auth-list>* must be a subset of the flags specified for *<auth-list>* with the **system-authority-out** command. The default is **none** (no protection authority flags may be present).

The syntax for the **port-authority-out** command is identical to that for the **system-authority-out** command, with the addition of a *<seg-list>*.

### *4.10.4 Setting the Basic-Security-Option Requirement*

For each segment, you can specify whether datagrams must contain a BSO (Basic Security Option). You use different commands to specify whether this option must be present for incoming datagrams and outgoing datagrams. Note that you can specify that BSO must be present in incoming datagrams on a specific segment only if you first specify BSO for outgoing datagrams on that segment.

#### *4.10.4.1 Outgoing Datagrams*

The **port-bso-reqd-xmit** command specifies whether outgoing (forwarded) datagrams transmitted on this segment must contain a BSO. If you specify **yes**, then unlabeled packets are not forwarded. If you specify **no**, then unlabeled packets are forwarded without modification (except that any security fields present are stripped). There is no corresponding system-wide command. The syntax for the **port-bso-reqd-xmit** command is:

```
security|sec <seg-list> port-bso-reqd-xmit|pbx
                yes|y|no|n
```

The default is **no**. If you specify **no**, the **port-bso-reqd-recv** parameter for that segment is forced to **no**.

#### *4.10.4.2 Incoming Datagrams*

The **port-bso-reqd-recv** command specifies whether incoming datagrams received on this segment must contain a BSO. If you specify **yes**, then unlabeled packets (ones that do not contain a BSO) are not forwarded. If you specify **no**, then unlabeled packets are assigned the security options specified by the **sec port-implicit-label** command (see Section 4.10.6.2 on page 107) and processed accordingly. There is no corresponding system-wide command.

The syntax is:

```
security|sec <seg-list> port-bso-reqd-recv|pbr
               yes|y|no|n
```

The default is **no**. To avoid configuration errors, the **port-bso-reqd-recv** parameter can be set to **yes** only if the **port-bso-reqd-xmit** parameter has already been set to **yes** for this segment.

## 4.10.5 Dealing with Unlabeled Datagrams

You can define how the PowerHub system handles datagrams that do not have a BSO label. For outgoing datagrams, you can use the **`port-def-authority-out`** command to set or clear a single Default-Protection-Authorities field. For incoming packets, you can assign a single classification level and Protection-Authority-Flag field using the **`port-implicit-label`** command. The syntax for these commands is shown in the following sections.

### 4.10.5.1 Setting or Clearing a Default-Protection-Authorities Field

The **`port-def-authority-out`** command sets or clears a single Default-Protection-Authorities field attached to unlabeled datagrams transmitted from the specified segment. This is not an RFC 1108 parameter but it gives you the ability to configure default security settings for outgoing packets. This field must be one of those specified for the **`port-authority-out`** parameter for the segment, which in turn must be present in the **`system-authority-out`** parameter.

The syntax for this command is:

```
security|sec <seg-list>
                port-def-authority-out|pdao
                none|<auth-list>
```

where:

*`<seg-list>`* Specifies the segment(s) for which you are defining the Default-Protection-Authorities field.

**`none|`***`<auth-list>`* Specifies the contents of the Default-Protection-Authorities field. You can specify **none** or a list containing any of the following:

**`genser|gs`**

**`siop-esi|se`**

**`sci|sc`**

**`nsa|ns`**

**`doe|do`**

You can specify only a single *`<flag-list>`* field. There is no corresponding system-wide command. The default is **`none`** (no protection authorities added).

#### 4.10.5.2 Assigning a Classification Level

The **port-implicit-label** command assigns a single classification level and a single Protection-Authority-Flag field to all unlabeled datagrams received on this segment. This parameter is meaningful only if the **port-bso-reqd-recv** parameter is set to **no**. There is no corresponding system-wide command.

The syntax for this command is:

```
security|sec <seg-list>
            port-implicit-label|pil <level>
            [none|<auth-list>]
```

The argument values are exactly the same as those for the **port-def-authority-out** command, described in the previous section.

### 4.10.6 Specifying the Classification Levels

Use the commands in this section to specify more restrictive maximum and minimum classification levels permitted in any datagram transmitted or received by a particular segment. These levels must fall within the range defined for the PowerHub system using the **system-level-max** and **system-level-min** commands.

#### 4.10.6.1 Maximum Classification Level

To specify the highest classification level permitted in any datagram transmitted or received on a particular segment or segments, use the **sec port-level-max** command:

```
security|sec <seg-list> port-level-max|plx
            <level>
```

The *<level>* must be less than or equal to the level specified with the **system-level-max** parameter and greater than or equal to **port-level-min**. The default is **unclassified**.

#### 4.10.6.2 Minimum Classification Level

To specify the lowest classification level permitted in any datagram transmitted or received on a particular segment or segments, use the **sec port-level-min** command:

```
security|sec <seg-list> port-level-min|pln
            <level>
```

The *<level>* must be greater than or equal to the level specified with the **system-level-min** parameter and less than or equal to **port-level-max**. The default is **unclassified**.

### 4.10.7 Stripping the Basic Security Option from Outgoing Datagrams

The **sec port-strip-bso** command enables you to remove the Basic Security Option from packets going out on the specified segment or segments. This feature is particularly useful if you are sending to a system that does not implement security labeling. There is no corresponding system-wide command.

The syntax for this command is:

```
security|sec <seg-list> port-strip-bso|psb
                 no|n|yes|y
```

This parameter is meaningful only if the **port-bso-reqd-xmit** parameter is set to **no**. The default is **no**.

# Part 2: Filtering

This part provides information on the commands you use to create IPX RIP and SAP filters for IPX, or zone and forwarding filters for AppleTalk. This part contains the following chapters:

Chapter 5: AppleTalk Filtering Commands

Describes filtering commands in the **atalk** subsystem.

Chapter 6: IPX Filtering Commands

Describes filtering commands in the **ipx** subsystem.

For information on other types of filters, see the following chapters in the *PowerHub Software Manual, V 2.6*:

| | |
|---|---|
| Bridge filter commands | Chapter 9. |
| TCP filter commands | Chapter 10. |
| IP filter commands | Chapter 11. |
| RIP filter commands | Chapter 12. |

# 5 AppleTalk Filtering Commands

AppleTalk Zone and NBP (Name Binder Protocol) filters provide security control over the server information sent and received by the AppleTalk networks associated with your PowerHub segments. Filters can perform the following functions:

- Send or block zone updates on specified PowerHub segment(s) that are sent from a particular network.
- Accept or discard a specific zone name received on a specific network.
- Report or hide a specific zone name in a specific network.
- Send or block NBP information on a specific segment.

## 5.1 ACCESSING THE IPX FILTER COMMANDS

All the commands in this chapter are in the **`atalk`** subsystem. To access the **`atalk`** subsystem, enter **`atalk`** at the runtime command prompt.

## 5.2 EXCLUSIVITY

AppleTalk filters of the same type (update, accept, report, and NBP forwarding filters) are mutually exclusive. If you define a filter that explicitly receives or sends specific information, all other information is implicitly discarded. For example, if you define an accept filter that explicitly accepts updates from a specific network, all other updates from that network are discarded. To accept additional updates from that network, you need to define additional filters. However, updates received from other networks are not affected.

If you need to secure access to just a few networks, it is generally easier to define filters that block or discard update information sent on or received just from those networks. All update information not explicitly blocked is forwarded. However, if your network requires tight security, you can define filters that explicitly allow only specific updates to be sent or received.

## 5.3 INPUT FILTERS AND OUTPUT FILTERS

When you define an AppleTalk zone filter, the items of information you supply are the network range and the segment number, or the zone name and segment number. For NBP filters, the items of information you supply are the zone name and segment number, or the AppleTalk object type and network range. The use of these arguments depends on whether you are defining an input filter or an output filter:

*Input filter* Operates on the receiving end of the report or update. When a network or a specific segment receives a report or update, input filters accept or reject information in the update. Zone accept filters are input filters.

*Output filter* Operates on the sending end of the report or update. Before an update or report is sent for an AppleTalk network on a specific segment, output filters report or discard entries in the update or report. Zone update and zone report filters are output filters.

## 5.4 APPLYING MULTIPLE FILTERS

Filters are applied in ascending numerical order (from the lowest filter number to the next highest). Therefore, filters should be defined in a “most important” to “least important” order. If you define more than one filter, the following rules determine how the filters are applied:

The filtering process accepts or discards packets when a filter finds the AppleTalk zone or network number that it is constructed to match. If a match is made, the filter performs its user-defined function. AppleTalk zone filters and NBP filters work in the following manner:

- If all filters are `report` or `send` filters and there is no match, the zone or NBP object is hidden or blocked.
- If all filters are `hide` or `block` filters and there is no match, the zone or NBP object is reported.
- If both `report` and `hide` filters, or `send` and `block` filters, are defined and there is no match, the zone is hidden. To change this behavior, define the last filter (filter number 128) as a report filter that matches all zones.

## 5.5 ZONE FILTER COMMANDS

Table 5–1 lists and describes the AppleTalk filter commands and their syntax. For each command, the management capability (root or monitor) is listed, as well as the section that contains additional information about the command.

**TABLE 5–1** AppleTalk zone filter commands.

| Command and Description | Capability† | See… |
|---|---|---|
| **zone-accept-filter\|zaf**<br>**add\|a** *<filnum>* **accept\|a \| discard\|d** *<rcv-nw-range>***\|all** *<zone-name>***\|***<br>**chng\|c** *<filnum>* **accept\|a \| discard\|d** **[***<rcv-nw-range>***\|all** *<zone-name>***\|*]**<br>**show\|s** *<filter-list>***\|all**<br>**del\|d** *<filter-list>***\|all**<br>Adds, changes, deletes, or shows AppleTalk zone accept filters. | R | 5.5.2 |
| **zone-report-filter\|zrf**<br>**add\|a** *<filnum>* **report\|r \| hide\|h** *<snd-nw-range>***\|all** *<zone-name>***\|***<br>**chng\|c** *<filnum>* **report\|r \| hide\|h** **[***<snd-nw-range>***\|all** *<zone-name>***\|*]**<br>**show\|s** *<filter-list>***\|all**<br>**del\|d** *<filter-list>***\|all**<br>Adds, displays, changes, or deletes AppleTalk zone report filters. | R | 5.5.3 |
| †R= Root, M= Monitor. | | |

**TABLE 5–1** (Continued) AppleTalk zone filter commands.

<table>
<tr><th>Command and Description</th><th>Capability†</th><th>See…</th></tr>
<tr><td>zone-update-filter|zuf<br>add|a &lt;filnum&gt; send|s | block|b<br>&lt;segment&gt; &lt;snd-nw-range&gt;<br>chng|c &lt;filnum&gt; send|s | block|b<br>[&lt;segment&gt; &lt;snd-nw-range&gt;]<br>show|s &lt;filter-list&gt;|all<br>del|d &lt;filter-list&gt;|all<br>Adds, displays, changes, or deletes AppleTalk zone update filters.</td><td>R</td><td>5.5.1</td></tr>
<tr><td colspan="3">†R= Root, M= Monitor.</td></tr>
</table>

The following sections describe the commands to add, display, change, and delete AppleTalk zone filters.

**NOTE**: Because zone and network information is exchanged when a zone or interface is configured, to ensure the proper functioning of your AppleTalk zone filters, we recommend the following setup procedures:

- Disable AppleTalk routing first on the PowerHub system. This step clears all previously learned zones (if any).
- Apply zone filters to your PowerHub system before adding any zones or interfaces. This step ensures that the filter for the appropriate zone is in place before any zones are learned on the specified hub.
- Re-enable AppleTalk routing. This step re-introduces the zone information except for the zone(s) that the filter was constructed to control.

## 5.5.1 Update Filter Commands

The following sections describe how to add, display, change, and delete zone update filters.

### 5.5.1.1 Adding an Update Filter

Use the **`zone-update-filter`** command to define and apply update filters. Here is the syntax for this command:

```
zone-update-filter|zuf add|a <filnum>
            send|s | block|b <segment> <snd-nw-range>
```

where:

*`<filnum>`* Specifies the filter number. Filter numbers must fall within the range 1 – 128.

**`send|s | block|b`** Specifies whether AppleTalk zone updates are sent or blocked on the specified network. The default is **`send.`**

*`<segment>`* Specifies the segment(s) on which updates are to be sent or blocked. Segments must fall within the range of 1 – 64.

*`<snd-nw-range>`* Specifies the network on which the updates are sent. You can only have one network range per PowerHub segment.

### 5.5.1.2 Displaying an Update Filter

Use the **`zone-update-filter show`** command to display currently defined zone update filters. Here is the syntax for this command:

```
zone-update-filter|zuf show|s <filter-list>|all
```

where:

*`<filter-list>`***`|all`** Specifies the filter number(s) for which you want to display the definition. Enter a filter number, a comma-separated list of filter numbers, or **`all`** for all filters. Filter numbers must fall within the range 1 – 128.

### 5.5.1.3 Changing an Update Filter

Use the **`zone-update-filter chng`** command to change the definition of an existing zone update filter. Here is the syntax for this command:

```
zone-update-filter|zuf chng|c <filnum> send|s | block|b
           [<segment> <snd-nw-range>]
```

The arguments for this command are the same as the arguments for the **`zone-update-filter add`** command. See Section 5.5.1.1 on page 115 for a description of each argument. Note that only the *`<filnum>`* and **`send|s | block|b`** arguments are required whereas the remaining arguments are optional. This syntax lets you easily change an existing filter from a “send” filter to a “block” filter (and vice versa), without needing to completely redefine the filter.

#### 5.5.1.4 Deleting an Update Filter

Use the **`zone-update-filter del`** command to delete an AppleTalk zone update filter. When you delete the filter, the filter number is no longer associated with the filter definition and the filter no longer applies to any PowerHub segments. Here is the syntax for this command:

```
zone-update-filter|zuf del|d <filter-list>|all
```

where:

| | |
|---|---|
| *`<filter-list>`*`|all` | Specifies the filter number(s) for which you want to delete the definitions. Enter a filter number, a comma-separated list of filter numbers, or **`all`** for all filters. Filter numbers must fall within the range 1 – 128. |

### 5.5.2 Accept Filter Commands

Zone accept filters give you control of the zone report information received on a network. Depending on how you define an accept filter, the PowerHub system accepts or discards a zone name or all zone names that are received in updates on a particular network or all networks.

Accept filters are input filters. You apply them to the network ranges on which Zone Information Protocol (ZIP) response packets are received. AppleTalk uses ZIP to learn and maintain the pairing of networks and zone names. ZIP response packets carry the network-to-zone name pairing that is required when a router in an AppleTalk network sends a ZIP request.

Because zone information is exchanged when an interface first is configured, apply zone accept filters before adding an interface.

The following sections describe how to add, display, change, and delete zone accept filters.

#### 5.5.2.1 Adding an Accept Filter

Use the **`zone-accept-filter add`** command to define AppleTalk zone accept filters. Here is the syntax for this command:

```
zone-accept-filter|zaf add|a <filnum> accept|a | discard|d
    <rcv-nw-range>|all <zone-name>|*
```

where:

| | |
|---|---|
| *`<filnum>`* | Specifies the filter number. Enter a number in the range 1 – 128. |
| **`accept|a | discard|d`** | Specifies whether updates for the specified zone name are accepted or discarded. |
| *`<rcv-nw-range>`*`|all` | Specifies that the filter is applied to zone updates received on the particular network specified by *`<rcv-nw-range>`*, or on all networks if **`all`** is specified. |
| *`<zone-name`*`|*`*`>`* | Specifies the AppleTalk zone that will be filtered. Specifying an asterisk (**`*`**) causes all zones to be filtered. |

### *5.5.2.2 Displaying an Accept Filter*

Use the **`zone-accept-filter show`** command to display the definitions for accept filters. Here is the syntax for this command:

```
zone-accept-filter|zaf show|s <filter-list>|all
```

where:

`<filter-list>|all` Specifies the filter(s). You can specify a filter number, a comma-separated list of filter numbers, or **`all`** for all filters. Filter numbers must fall in the range 1 – 128.

### *5.5.2.3 Changing an Accept Filter*

Use the **`zone-accept-filter chng`** command to change the definitions of an existing accept filter. Here is the syntax for this command:

```
zone-accept-filter|zaf chng|c <filnum>
     accept|a | discard|d [<rcv-nw-range>|all <zone-name>|*]
```

The arguments for this command are the same as the arguments for the **`zone-accept-filter add`** command. See Section 5.5.2.1 on page 116 for a description of each argument. Note that only the *`<filnum>`* and **`accept|a | discard|d`** arguments are required whereas the remaining arguments are optional.

### *5.5.2.4 Deleting an Accept Filter*

Use the **`zone-accept-filter del`** command to delete the definitions of an accept filter. Here is the syntax for this command:

```
zone-accept-filter|zaf del|d <filter-list>|all
```

where:

`<filter-list>|all` Specifies the filter(s) you want to delete. You can specify a filter number, a comma-separated list of filter numbers, or **`all`** for all filters. Filter numbers must be within the range 1 – 128.

## 5.5.3 Report Filter Commands

Zone report filters give you the same name-by-name level of control as accept filters, except report filters control the information sent out by the PowerHub system, instead of controlling the information received by the PowerHub system.

Like update filters, report filters are output filters. They report or hide zones in ZIP responses generated and sent by the PowerHub system. Report filters allow you to specify a particular zone name that you want to report or hide. Apply them to the networks on which zone reports for a specific zone name are sent.

The following sections describe how to add, display, change, and delete zone report filters.

### 5.5.3.1 Adding a Report Filter

Use the **zone-report-filter add** command to create an AppleTalk zone report filter. Here is the syntax for this command:

```
zone-report-filter|zrf add|a <filnum>
    report|r | hide|h <snd-nw-range>|all <zone-name>|*
```

where:

| | |
|---|---|
| *<filnum>* | Specifies the filter number. Filters must fall within the range 1 – 128. |
| **report\|r \| hide\|h** | Specifies whether information about the specified zone is to be reported or hidden. |
| *<snd-nw-range>*\|**all** | Specifies the network sending the zone reports. If you specify **all**, the zone is filtered on all networks. |
| *<zone-name>*\|***** | Specifies the AppleTalk zone to be filtered; specifying an asterisk (*****) causes all zones to be filtered. |

### 5.5.3.2 Displaying a Report Filter

Use the **zone-report-filter show** command to display the definitions for report filters. Here is the syntax for this command:

```
zone-report-filter|zrf show|s <filter-list>|all
```

where:

| | |
|---|---|
| *<filter-list>*\|**all** | Specifies the filter(s). You can specify a filter number, a comma-separated list of filter numbers, or **all** for all filters. Filter numbers must fall within the range 1 – 128. |

### 5.5.3.3 Changing a Report Filter

Use the **zone-report-filter chng** command to change the definitions of an existing report filter. Here is the syntax for this command:

```
zone-report-filter|zrf chng|c <filnum>
   report|r | hide|h  [<snd-nw-range>|all <zone-name>|*]
```

The arguments for this command are the same as the arguments for the **zone-report-filter add** command. See Section 5.5.3.1 on page 118 for a description of each argument. Note that only the *`<filnum>`* and **`report|r | hide|h`** arguments are required whereas the remaining arguments are optional. This syntax lets you easily change an existing filter from a "report" filter to a "hide" filter (and vice versa), without needing to completely redefine the filter.

### *5.5.3.4 Deleting a Report Filter*

Use the **`zone-report-filter del`** command to delete the definition from a report filter number. Here is the syntax for this command:

**`zone-report-filter|zrf del|d`** *`<filter-list>`***`|all`**

where:

*`<filter-list>`***`|all`** Specifies the filter(s) you want to delete. You can specify a filter number, a comma-separated list of filter numbers, or **`all`** for all filters. Filter numbers must fall within the range 1 – 128.

## *5.6 NAME-BINDING-PROTOCOL FILTER COMMANDS*

AppleTalk NBP filters allow you to control the ability of the AppleTalk network to bind names and network services together. Using NBP filters, you can hide or report AppleTalk routers and hide zone names from certain types of NBP packets.

NBP filters apply to a zone name or network object and PowerHub segment or network range.

When the PowerHub system receives an NBP request, the PowerHub system can do two things with the request:

- If the PowerHub is directly attached to the device on which the requested service type exists, the PowerHub forwards an NBP Lookup request to the device.
- If the PowerHub system is not directly attached to a device that contains the requested service type, the PowerHub forwards an NBP Forward request to another device.

Thus, when the PowerHub system receives an NBP request for a service type, the hub sends the Lookup or Forward packet on the segment that contains the zone where the requested service resides.

For example, if NBP was looking for a service type which resided in the zone `"accounting"` on segment `5,` you could set up a filter that would allow or disallow the PowerHub system to send NBP requests to that zone. The following example shows one way to set up this filter.

```
1:PowerHub# nbp-fwd-filter add 1 block 5 accounting
Okay
```

This command applies filter number 1, which is a "block" filter, to PowerHub segment 5. This filter ensures that the PowerHub system will not forward any NBP request packets to the zone "`accounting`" on segment 5.

Table 5–2 lists and describes the AppleTalk Name Binding Protocol filter commands and their syntax. For each command, the management capability (root or monitor) is listed, as well as the section that contains additional information about the command.

**TABLE 5–2** AppleTalk Name Binding Protocol Filter Commands.

<table>
<tr><th>Command and Description</th><th>Capability†</th><th>See…</th></tr>
<tr><td><b>nbp-fwd-filter|nff</b><br><b>add|a</b> <i><filnum></i> <b>send|s | block|b</b><br><i><segment></i> <i><zone-name></i><br><b>chng|c</b> <i><filnum></i> <b>send|s | block|b</b><br><b>[</b><i><segment></i> <i><zone-name></i><b>]</b><br><b>show|s</b> <i><filter-list></i><b>|all</b><br><b>del|d</b> <i><filter-list></i><b>|all</b><br>Adds, displays, changes, or deletes Name Binding Protocol forwarding filters.</td><td>R</td><td>5.6.1</td></tr>
<tr><td colspan="3">†R= Root, M= Monitor.</td></tr>
</table>

## 5.6.1 Forwarding Filters

NBP forwarding filters are output filters; they typically block the datastream exiting the PowerHub system. They are global filters, meaning that they send or block all NBP Lookup requests exiting the hub on a particular segment going to a specified zone(s).

You can apply forwarding filters to a PowerHub segment to permit or prevent NBP Forward or Lookup requests from going out on that segment. A forwarding filter is assigned to a PowerHub segment, and you define the zone name to which you want NBP requests sent or blocked.

The following sections describe how to add, display, change, and delete NBP forwarding filters.

### 5.6.1.1 Adding a Forwarding Filter

To add a forwarding filter, issue the **nbp-fwd-filter add** command. Here is the syntax of this command:

```
nbp-fwd-filter|nff add|a <filnum>
      send|s | block|b <segment> <zone-name>
```

where:

*<filnum>* Specifies the filter number; you can enter a filter number, a comma-separated list of filters. Filter numbers must fall within the range 1 – 128.

| | |
|---|---|
| **send\|s \| block\|b** | Specifies whether you want the PowerHub system to send or block an NBP request. |
| *<segment>* | Indicates the segment on which the NBP Lookup request is sent or blocked. |
| *<zone-name>* | Specifies the AppleTalk zone which will receive or not receive NBP Lookup requests. |

### 5.6.1.2 Displaying a Forwarding Filter

To display an NBP filter, issue the **nbp-fwd-filter show** command. Here is the syntax for this command:

```
nbp-fwd-filter|nff show|s <filter-list>
```

where:

| | |
|---|---|
| *<filter-list>* | Specifies the number of the filter(s); you can specify a comma-separated list of filters. Filter numbers must fall within the range 1 – 128. |

### 5.6.1.3 Changing a Forwarding Filter

To change an existing NBP forwarding filter, issue the **nbp-fwd-filter chng** command. Here is the syntax for this command:

```
nbp-fwd-filter|nff chng|c <filnum> send|s | block|b
      [<segment> <zone-name>]
```

The arguments for this command are the same as the arguments for the **nbp-fwd-filter add** command. See Section 5.6.1.1 on page 120 for a description of each argument. Note that only the *<filnum>* and **send|s | block|b** arguments are required whereas the remaining arguments are optional. This syntax lets you easily change an existing filter from a “send” filter to a “block” filter (and vice versa), without needing to completely redefine the filter.

### 5.6.1.4 Deleting a Forwarding Filter

To delete a forwarding filter, issue the **nbp-fwd-filter del** command. Here is the syntax for this command:

```
nbp-fwd-filter|nff del|d <filter-list>
```

where:

| | |
|---|---|
| *<filter-list>* | Specifies the number of the filter(s); you can specify a filter number or a comma-separated list of filters. Filter numbers must fall within the range 1 – 128. |

# 6 IPX Filtering Commands

IPX RIP (Routing Information Protocol) and SAP (Service Advertising Protocol) filters give you security control over the route and server information sent and received by the IPX networks associated with your PowerHub segments. Depending on the level of security you want, you can:

- Block a segment from sending RIP or SAP updates for a particular network. (See Section 6.5 on page 132.)
- Block or accept route or server information received on a specific network. (See Section 6.6 on page 134.)
- Block or allow route or server information to be sent on a specific network. (See Section 6.7 on page 137.)
- Block servers from being reported in response to an IPX “Get Nearest Server” request. (See Section 6.7.2 on page 138.)

IPX RIP filters let you restrict connectivity to IPX networks by selectively controlling the routes that are reported or accepted in RIP updates.

IPX SAP filters let you restrict connectivity to IPX servers by controlling the receipt and transmission of SAP updates. Using SAP filters, you can “hide” secured servers from workstations that try to connect to them.

All filters apply to a specific network number or servers on a specific segment.

This chapter describes how each type of filter works and describes the commands you use to create, display, change, or delete the filters.

## 6.1 ACCESSING THE IPX FILTER COMMANDS

The IPX filter commands are located in the **ipx** subsystem. To access the **ipx** subsystem, issue the following command at the runtime command prompt:

```
ipx
```

## 6.2 IPX FILTER COMMANDS

Table 6–1 lists and describes the IPX filter commands and their syntax. For each command, the management capability (root or monitor) is listed, as well as the section that contains additional information about the command.

**TABLE 6–1** IPX filter commands.

| Command and Description | Capability* | See… |
|---|---|---|
| **rip-accept-filter\|raf** | R | 6.6 |
| **add\|a** *<filnum>* **accept\|a \| discard\|d**<br>*<network> <rcv-if-nw>***\|all** | | |
| **chng\|c** *<filnum>* **accept\|a \| discard\|d**<br>**[***<network> <rcv-if-nw>***\|all]** | | |
| **del\|d** *<filter-list>***\|all** | | |
| **show\|s** *<filter-list>***\|all** | R or M # | |
| Adds, displays, changes, or deletes RIP accept filters. | | |
| **rip-report-filter\|rrf** | R | 6.7 |
| **add\|a** *<filnum>* **report\|r \| hide\|h**<br>*<network> <snd-if-nw>***\|all** | | |
| **chng\|c** *<filnum>* **report\|r \| hide\|h**<br>**[***<network> <snd-if-nw>***\|all]** | | |
| **del\|d** *<filter-list>***\|all** | | |
| **show\|s** *<filter-list>***\|all** | R or M # | |
| Adds, displays, changes, or deletes RIP report filters. | | |

*****R= Root, M=Monitor.

#R= display and manipulate, M= display only.

**TABLE 6–1** (Continued) IPX filter commands.

<table>
<tr><th>Command and Description</th><th>Capability*</th><th>See…</th></tr>
<tr><td>rip-update-filter|ruf<br>add|a <filnum><br>send|s | block|b <seg> <network><br>chng|c <filnum> send|s | block|b<br>[<seg> <network>]<br>del|d <filter-list>|all<br>show|s <filter-list>|all<br>Adds, displays, changes, or deletes RIP update filters.</td><td>R<br>R or M #</td><td>6.5</td></tr>
<tr><td>sap-accept-filter|saf<br>add|a <filnum> accept|a | discard|d<br><svr-type> <svr-name>|* <rcv-if-nw>|all<br>chng|c <filnum> accept|a | discard|d<br>[<svr-type> <svr-name>|* <rcv-if-nw>|all]<br>del|d <filter-list>|all<br>show|s <filter-list>|all [-f]<br>Adds, displays, changes, or deletes SAP accept filters.</td><td>R<br>R or M #</td><td>6.6</td></tr>
<tr><td>sap-report-filter|srf add|a <filnum><br>report|r|hide|h |hide-nearest|hn<br><svr-type> <svr-name>|* <snd-if-nw>|all<br>chng|c <filnum><br>report|r|hide|h|hide-nearest|hn<br>[<svr-type> <svr-name>|* <snd-if-nw>|all]<br>del|d <filter-list>|all<br>show|s <filter-list>|all [-f]<br>Adds, displays, changes, or deletes SAP report filters.</td><td>R<br>R or M #</td><td>6.5</td></tr>
<tr><td colspan="3">*R= Root, M=Monitor.<br>#R= display and manipulate, M= display only.</td></tr>
</table>

**TABLE 6–1** (Continued) IPX filter commands.

| Command and Description | Capability* | See… |
|---|---|---|
| **sap-update-filter\|suf add\|a** *<filnum>*<br>**send\|s\|block\|b** *<seg> <network>* | R | 6.5 |
| **chng\|c** *<filnum>* **send\|s\|block\|b**<br>**[***<seg> <network>***]** | | |
| **del\|d** *<filter-list>***\|all** | | |
| **show\|s** *<filter-list>***\|all** | R or M # | |
| Adds, displays, changes, or deletes SAP update filters. | | |
| ***R= Root, M=Monitor.**<br>#R= display and manipulate, M= display only. | | |

## 6.3 HOW IPX RIP AND SAP FILTERS WORK

IPX RIP filters control the route information sent or received by a specific IPX network number on a specific segment. Similarly, IPX SAP filters control the server information sent or received.

### 6.3.1 Exclusivity

Filters of the same type (update, accept, or report) are mutually exclusive. If you define a filter that explicitly receives or sends specific information, all other information is implicitly discarded. For example, if you define a RIP accept filter that explicitly accepts RIP updates from a specific IPX network, all other RIP updates are discarded. To accept additional RIP updates, you need to define additional filters.

If you need to secure access to just a few networks or servers, it’s generally easier to define filters that block or discard update information sent by those networks or servers. All update information not explicitly blocked is forwarded. However, if your network requires tight security, you can define filters that explicitly allow only specific updates to be sent or received.

### 6.3.2 Input Filters and Output Filters

When you define an IPX RIP or SAP filter, one of the items of information you supply is the segment number. The number you supply varies depending upon whether you are defining an input filter or an output filter:

*Input filter* Operates on the receiving end of the RIP or SAP update. When an IPX network on a specific segment receives a RIP or SAP update, input filters accept or reject information in the update. Accept filters are input filters.

*Output filter* Operates on the sending end of the RIP or SAP update. Before the update is sent for an IPX network on a specific segment, output filters report or discard entries in the update. Update and report filters are output filters.

### 6.3.3 Types of Control

The IPX RIP and SAP filters give you different types of control. Depending upon the types of filters you define, you can filter according to the following:

Network and interface combination

Accept and Report filters let you filter according to specific networks on specific interfaces.

Using accept filters, you can selectively accept or discard updates sent from specific interfaces on a specific network.

Using report filters, you can selectively send or block updates from a specific interface on a specific network.

Segment and interface combination

Update filters let you filter according to specific interfaces on specific segments. You can selectively send or block updates to a specific interface on a specific segment.

## 6.4 EXAMPLES OF FILTERS

This section gives examples of each type of IPX RIP and SAP filter: update, accept, and report filters. All of these examples are based on the example IPX network shown in Figure 6–1.

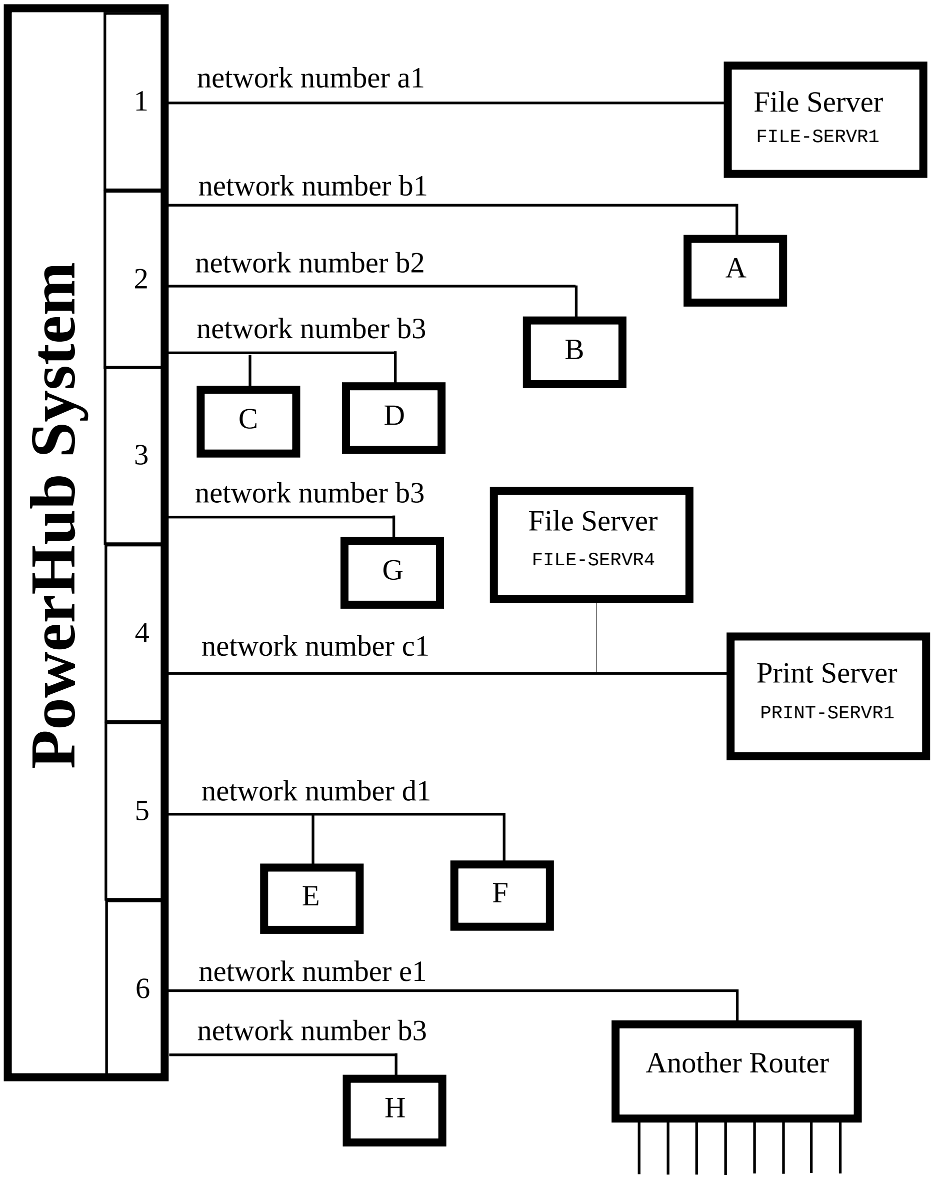


**FIGURE 6–1** Example of an IPX network.

In Figure 6–1, the numbered boxes inside the PowerHub systems represent segments on which IPX networks have been defined. This example assumes that IPX network numbers have been correctly defined on the PowerHub system, on the servers, and on the end nodes. For simplicity, this example also assumes that all devices are using the same encapsulation type.

The following sections give examples of how each type of filter might be used in the configuration shown in Figure 6–1 on page 128. The examples include the commands used to create the filters.

## 6.4.1 Update Filters

Update filters cause the PowerHub system sending the RIP or SAP updates to send or not send updates to an entire network on a specific segment. Use update filters if you want to establish security at a broad level. Update filters are output filters. You apply them to the networks and segments to prevent all RIP or SAP updates from being sent on a network/segment.

The following sections give examples of RIP and SAP update filters.

### *6.4.1.1 RIP Update Filter*

Suppose you want to define a filter that prevents the PowerHub system from sending RIP updates to devices connected to network b3 on segment 2. Note that you must specify both a network and a segment because the PowerHub’s Virtual LAN capability allows more than one IPX network to be defined on a given segment, and more than one segment to contain nodes of a given IPX network.

You might use the **`rip–update-filter add`** command to create a filter as follows:

```
1:PowerHub# rip-update-filter add 1 block 2 b3
Okay
```

This command creates (adds) RIP update filter number 1, which blocks RIP updates from being sent to devices on segment 2 that belong to network b3. These devices will not learn any IPX routes from the hub.

### *6.4.1.2 SAP Update Filter*

Here is an example of a SAP update filter, using the same configuration shown in Figure 6–1. Suppose you want to block server information from being sent to networks b1 and b2 on segment 2. You can issue a command such as the following to define a filter to do this:

```
1:PowerHub# sap-update-filter add 2 send 2 b3
Okay
```

This command creates SAP update filter number 2, which allows all server information to be sent to network b3 on segment 2. However, updates are not sent on networks b1 or b2 on segment 2. Consequently, devices on these networks will not get any server information from the PowerHub system.

## 6.4.2 Accept Filters

RIP and SAP accept filters control the route and server information that the PowerHub system receives from an IPX network. Depending on how you define an accept filter, it accepts or discards updates received on a single network or for all networks:

- For RIP updates, you can accept or discard specific route information received from a specific network. The network can be defined on a single PowerHub segment or multiple PowerHub segments (VLAN).
- For SAP updates, you can accept or discard updates about a specific server, or about all servers of a specific type.

Unlike update filters, accept filters are input filters. You apply them to the networks for which the RIP or SAP updates are received.

The following sections give examples of RIP and SAP accept filters.

### 6.4.2.1 RIP Accept Filter

Suppose you want to define a filter that allows all route information to be accepted from network b1, with the exception of routes to network 1234. (Note that network 1234 is not directly attached to the PowerHub system.) You can define a filter such as this using the following command:

```
1:PowerHub# rip-accept-filter add 1 discard 1234 b1
Okay
```

This command specifies that route information for network 1234 is to be discarded when received on network b1. The PowerHub software accepts all other route information that it receives on this network. To discard additional routes, you would need to define additional filters.

### 6.4.2.2 SAP Accept Filter

Suppose you want to ensure that the only server on network c1 that can be accessed from other networks is the print server print-servr1. You can define a SAP accept filter as follows:

```
1:PowerHub# sap-accept-filter add 1 accept print-servr print-servr1 c1
Okay
```

This command specifies that SAP information about the server named “`print-servr1`,” which is a server of the type PRINT-SERVR, should be accepted from network c1, but no other SAP information should be accepted from that network. If you add additional servers to this IPX network, this filter throws away SAP updates from the other servers to ensure that no clients on other networks can discover the new servers.

## 6.4.3 Report Filters

RIP and SAP report filters give you the same level of control as RIP accept filters, except report filters control the information sent out on a network, instead of controlling the information received on a network.

Like update filters, report filters are output filters. You apply them to the networks on which the RIP or SAP updates are sent.

The following sections give examples of RIP and SAP report filters.

### 6.4.3.1 RIP Report Filter

Suppose you want to “hide” the routes to all networks except b1 and b2 from devices on network d1. You can accomplish this by defining the following RIP report filters:

```
1:PowerHub# rip-report-filter add 1 report b1 d1
Okay
1:PowerHub# rip-report-filter add 2 report b2 d1
Okay
```

Because a report filter automatically hides routes not explicitly specified in the filter definition, these two filters ensure that route information for b1 and b2 only is visible to devices on network d1.

### 6.4.3.2 SAP Report Filter

Suppose that you want to hide the file server named “`file-server1`” on network a1 from networks b1 and b2, but you want to allow it to be reported to network b3. You can do this by creating the following SAP report filters:

```
1:PowerHub# sap-report-filter add 1 hide file-server file-servr1 b1
Okay
2:PowerHub# sap-report-filter add 2 hide file-server file-servr1 b2
Okay
3:PowerHub# sap-report-filter add 3 report file-server file-servr1 b3
Okay
```

These filters prevent the file server “`file-server1`” from being reported to networks b1 and b2, but report all other servers to these networks on this segment (unless you define additional filters to hide those servers). File server “`file-server1`” *is* reported to network b3, but other file servers are not reported to this network, unless additional filters are defined to allow the servers to be reported.

You also can define a SAP Report filter to hide servers from a Get Nearest Server request. The server is hidden from the workstations on the specified networks, but the server is reported to other routers on the network. Here is an example of the definition of such a filter:

```
1:PowerHub#  srf add 4 hide-nearest file-server file-servr1 all
Okay
```

This example creates SAP report filter 4 to hide the file server called "`file-server1`" from being reported in response to Get Nearest Server requests issued from any network.

# 6.5 UPDATE FILTER COMMANDS

The following sections describe the commands to add, display, change, and delete IPX RIP and SAP update filters.

## 6.5.1 Adding a RIP Update Filter

Use the **`rip-update-filter`** command to define and apply RIP update filters. Here is the syntax for this command:

```
rip-update-filter|ruf add|a
      <filnum> send|s|block|b <seg> <network>
```

where:

*`<filnum>`* Specifies the filter number, in the range 1–128.

**`send|s|block|b`** Specifies whether all IPX RIP updates are sent or blocked on the specified IPX network.

*`<seg>`* Specifies the segment to which you are applying the filter.

*`<network>`* Specifies the IPX interface to which you are applying the filter.

## 6.5.2 Adding a SAP Update Filter

Use the **`sap-update-filter add`** command to create a SAP update filter. The syntax for this command is similar to the syntax for the **`rip-report-filter add`** command; the only difference is, instead of specifying a network number, you specify a server type and server name. Here is the syntax for this command:

```
sap-update-filter|suf add|a
      <filnum> send|s|block|b <seg> <network>
```

where:

*`<filnum>`* Specifies the filter number, in the range 1–128.

**`send|s|block|b`** Specifies whether IPX SAP updates are sent or blocked on the specified IPX network.

*`<seg>`* Specifies the segment to which you are applying the filter.

*`<network>`* Specifies the IPX interface to which you are applying the filter.

### 6.5.3 Displaying a RIP Update Filter

Use the **rip-update-filter show** command to display currently defined IPX RIP update filters. Here is the syntax for this command:

**rip-update-filter|ruf show|s** *<filter-list>***|all**

where:

| | |
|---|---|
| *<filter-list>***\|all** | Specifies the filter numbers for which you want to display the definition. You can specify a single filter number, a comma-separated list of filter numbers, or **all** for all filters. |

### 6.5.4 Displaying a SAP Update Filter

Use the **sap-update-filter show** command to display the current definitions for SAP filters. Here is the syntax for this command:

**sap-update-filter|suf show|s** *<filter-list>***|all**

where:

| | |
|---|---|
| *<filter-list>***\|all** | Specifies the filter(s) for which you want to display definitions. You can specify a single filter number, a comma-separated list of filter numbers, or **all** for all filter definitions. |

### 6.5.5 Changing a RIP Update Filter

Use the **rip-update-filter chng** command to change the definition of an existing IPX RIP update filter. Here is the syntax for this command:

```
rip-update-filter|ruf chng|c
        <filnum> send|s|block|b [<seg> <network>]
```

The arguments for this command are the same as the arguments for the **rip-update-filter add** command. See Section 6.5.1 on page 132 for a description of each argument. Note that only the *<filnum>* and **send|s|block|b** arguments are required whereas the remaining arguments are optional. This lets you easily change an existing filter from a send filter to a block filter, without needing to completely redefine the filter.

### 6.5.6 Changing a SAP Update Filter

Use the **sap-update-filter chng** command to change the definition of a currently defined SAP update filter. Here is the syntax for this command:

```
sap-update-filter|suf chng|c
        <filnum> send|s|block|b [<seg> <network>]
```

The arguments for this command are the same as the arguments for the **sap-update-filter add** command. See Section 6.5.2 on page 132 for a description of each argument. Note that only the *<filnum>* and **send|s|block|b** arguments are required whereas the remaining arguments are optional. This lets you easily change an existing filter from a send filter to a block filter, without needing to completely redefine the filter.

### 6.5.7 Deleting a RIP Update Filter

Use the **rip-update-filter del** command to delete an IPX RIP update filter. When you delete the filter, the filter definition is no longer associated with the filter number and the filter no longer applies to any PowerHub segments. Here is the syntax for this command:

```
rip-update-filter|ruf del|d <filter-list>|all
```

where:

| | |
|---|---|
| *<filter-list>*\|**all** | Specifies the filter numbers from which you want to delete the definitions. You can specify a single filter number, a comma-separated list of filter numbers, or **all** for all filters. |

### 6.5.8 Deleting a SAP Update Filter

Use the **sap-update-filter del** command to delete the definition associated with a SAP update filter number. When you delete the filter definition, the filter is no longer associated with the filter number and the filter no longer applies to any PowerHub segments or IPX networks. Here is the syntax for this command:

```
sap-update-filter|suf del|d <filter-list>|all
```

where:

| | |
|---|---|
| *<filter-list>*\|**all** | Specifies the filter(s) you want to delete. You can specify a single filter number, a comma-separated list of filter numbers, or **all** for all filter definitions. |

## 6.6 ACCEPT FILTER COMMANDS

The following sections describe the commands to add, display, change, and delete IPX RIP and SAP accept filters.

### 6.6.1 Adding a RIP Accept Filter

Use the **rip-accept-filter add** command to define IPX RIP accept filters. Here is the syntax for this command:

```
rip-accept-filter|raf add|a
       <filnum> accept|a|discard|d <network> <rcv-if-nw>|all
```

where:

| | |
|---|---|
| *<filnum>* | Specifies the filter number. Enter a number in the range 1–128. |
| **accept\|a\|discard\|d** | Specifies whether RIP updates for the specified network number are accepted or discarded. |
| *<network>* | Specifies the IPX network number whose routes are to be accepted or discarded. |

*<rcv-if-nw>*|**all**

Specifies the network interface number on which route reports are received. If you specify an IPX network number, route reports originating from a device on that network are filtered. If you specify **all**, route reports originating from all networks on the specified segment are filtered.

### 6.6.2 Adding a SAP Accept Filter

Use the **sap-accept-filter add** command to create a SAP accept filter. The syntax for this command is similar to the syntax for the **rip-accept-filter add** command; the only difference is, instead of specifying a network number whose routes are to be filtered, you specify the server type and server name whose service advertisements are to be filtered. Here is the syntax for this command:

```
sap-accept-filter|saf add|a
        <filnum> accept|a|discard|d <svr-type> <svr-name>|*
        <rcv-if-nw>|all
```

where:

*<filnum>*

Specifies the filter number. Enter a number from 1–128.

**accept|a|discard|d**

Specifies whether SAP information for the specified server name and type is to be accepted or discarded.

*<svr-type>*

Specifies the server type. You can enter the mnemonic value or the number.

| Mnemonic | Hex equivalent |
|---|---|
| PRINT-QUEUE | 0003 |
| FILE-SERVER | 0004 |
| JOB-SERVER | 0005 |
| PRINT-SERVR | 0007 |
| ARCHIVE-SVR | 0009 |
| REM-BRIDGE | 0024 |
| ADVRT-PRINT | 0047 |

The numbers shown are hex values for 16-bit numbers. If you use the number for the server type, enter it as shown above.

*<svr-name>*|*

Specifies the name of the server. You can specify an individual server or enter * to apply the filter to all servers of the specified type.

*<rcv-if-nw>*|**all**

Specifies the network number on which the server report is received. You can enter a specific network number or **all** for all networks on the specified segment. Specifying **all** causes the server to be filtered in reports received from all networks configured on the specified segment.

### 6.6.3 Displaying a RIP Accept Filter

Use the **rip-accept-filter show** command to display the definitions for RIP accept filters. Here is the syntax for this command:

**rip-accept-filter|raf show|s** *<filter-list>***|all**

where:

| | |
|---|---|
| *<filter-list>***\|all** | Specifies the filter(s). You can specify a single filter number, a comma-separated list of filter numbers, or **all** for all filters. |

### 6.6.4 Displaying a SAP Accept Filter

Use the **sap-accept-filter show** to display the currently defined SAP accept filters. Here is the syntax for this command:

**sap-accept-filter|saf show|s** *<filter-list>***|all [-f]**

where:

| | |
|---|---|
| *<filter-list>***\|all** | Specifies the filter(s) for which you want to display definitions. You can specify a single filter number, a comma-separated list of filter numbers, or **all** for all filter definitions. |
| **-f** | Displays the full name of the server. By default only the first 39 characters of the server name are displayed. |

### 6.6.5 Changing a RIP Accept Filter

Use the **rip-accept-filter chng** command to change the definitions of an existing RIP accept filter. Here is the syntax for this command:

**rip-accept-filter|raf chng|c**

*<filnum>* **accept|a|discard|d [***<network> <rcv-if-nw>***|all]**

The arguments for this command are the same as the arguments for the **rip-accept-filter add** command. See Section 6.6.1 on page 134 for a description of each argument. Note that only the *<filnum>* and **accept|a|discard|d** arguments are required whereas the remaining arguments are optional. This lets you easily change an existing filter from an accept filter to a discard filter, without needing to completely redefine the filter.

### 6.6.6 Changing a SAP Accept Filter

Use the **sap-accept-filter chng** to change the definition of an existing SAP accept filter. Here is the syntax for this command:

**sap-accept-filter|saf chng|c**

*<filnum>* **accept|a|discard|d**

**[***<svr-type> <svr-name>***|*** *<rcv-if-nw>***|all]**

The arguments for this command are the same as the arguments for the **sap-accept-filter add** command. See Section 6.6.2 on page 135 for a description of each argument. Note that only the *<filnum>* and **accept|a|discard|d** arguments are required whereas the remaining arguments are optional. This lets you easily change an existing filter from an accept filter to a discard filter, without needing to completely redefine the filter.

### 6.6.7 Deleting a RIP Accept Filter

Use the **rip-accept-filter del** command to delete the definitions of a RIP accept filter. Here is the syntax for this command:

```
rip-accept-filter|raf del|d <filter-list>|all
```

where:

| | |
|---|---|
| *<filter-list>*\|**all** | Specifies the filter(s) you want to delete. You can specify a single filter number, a comma-separated list of filter numbers, or **all** for all filters. |

### 6.6.8 Deleting a SAP Accept Filter

Use the **sap-accept-filter del** to delete the definition of a SAP accept filter. The syntax for this command is similar to the syntax for the **rip-accept-filter del** command:

```
sap-accept-filter|saf del|d <filter-list>|all
```

where:

| | |
|---|---|
| *<filter-list>*\|**all** | Specifies the filter(s) you want to delete. You can specify a single filter number, a comma-separated list of filter numbers, or **all** for all filter definitions. |

## 6.7 REPORT FILTER COMMANDS

The following sections describe the commands to add, display, change, and delete IPX RIP and SAP report filters.

### 6.7.1 Adding a RIP Report Filter

Use the **rip-report-filter add** to create an IPX RIP report filter. Here is the syntax for this command:

```
rip-report-filter|rrf add|a
      <filnum> report|r|hide|h <network> <snd-if-nw>|all
```

where:

| | |
|---|---|
| *<filnum>* | Specifies the filter number. Enter a number in the range 1–128. |
| **report\|r\|hide\|h** | Specifies whether route information about the specified IPX network is to be reported or hidden. |

*<network>* Specifies the IPX network to be filtered:

- If you specified **report**, this network number is reported and all others are hidden (unless reported by other filters) from the RIP report.
- If you specified **hide**, this network is omitted from the RIP report and all others (unless hidden by other filters) are reported.

*<snd-if-nw>***|all** Specifies the network on which the filtered RIP reports are sent. If you specify **all**, the filter applies to all networks configured on the specified sending segment.

### 6.7.2 Adding a SAP Report Filter

Use the **sap-report-filter add** command to create a SAP report filter. The syntax for this command is similar to the syntax for the **rip-report-filter add** command; the only difference is, instead of specifying a network number whose routes are to be filtered, you specify the server type and server name whose service advertisements are to be filtered. Here is the syntax for this command:

```
sap-report-filter|srf add|a   <filnum>
       report|r|hide|h|hide-nearest|hn
       <svr-type> <svr-name>|* <snd-if-nw>|all
```

where:

*<filnum>* Specifies the filter number. Enter a number in the range 1–128.

**report|r|hide|h|hide-nearest|hn**

Specifies whether the filter is to report or hide server information. If you specify **hide-nearest**, the server you specify is hidden from being reported in response to IPX Get Nearest Server requests. Thus, the server is hidden from workstations on the specified networks, but is still reported to other routers on the network.

*<svr-type>* Specifies the server type. You can enter the mnemonic value or the number:

| **Mnemonic** | **Hex equivalent** |
|---|---|
| PRINT-QUEUE | 0003 |
| FILE-SERVER | 0004 |
| JOB-SERVER | 0005 |
| PRINT-SERVR | 0007 |
| ARCHIVE-SVR | 0009 |
| REM-BRIDGE | 0024 |
| ADVRT-PRINT | 0047 |

The numbers shown are hex values for 16-bit numbers. If you use the number for the server type, enter it as shown above.

*<svr-name>*|* Specifies the name of the server. You can specify an individual server or enter * to apply the filter to all servers of the specified type.

*<snd-if-nw>*|**all** Specifies the network number on which the server report is sent. You can enter a specific network number or **all** for all networks. Specifying **all** causes the server to be filtered in reports sent to all networks configured on the specified segment.

### 6.7.3 Displaying a RIP Report Filter

Use the **rip-report-filter show** command to display the definitions for RIP update filters. Here is the syntax for this command:

```
rip-report-filter|rrf show|s <filter-list>|all
```

where:

*<filter-list>*|**all** Specifies the filter(s). You can specify a single filter number, a comma-separated list of filter numbers, or **all** for all filters.

### 6.7.4 Displaying a SAP Report Filter

Use the **sap-report-filter show** command to display currently defined SAP report filters. Here is the syntax for this command:

```
sap-report-filter|srf show|s <filter-list>|all [-f]
```

where:

*<filter-list>*|**all** Specifies the filter(s) for which you want to display definitions. You can specify a single filter number, a comma-separated list of filter numbers, or **all** for all filter definitions.

**-f** Displays the full name of the server. By default only the first 39 characters of the server name are displayed.

### 6.7.5 Changing a RIP Report Filter

Use the **rip-report-filter chng** command to change the definitions of an existing RIP accept filter. Here is the syntax for this command:

```
rip-report-filter|rrf chng|c
       <filnum> report|r|hide|h [<network> <snd-if-nw>|all]
```

The arguments for this command are the same as the arguments for the **rip-report-filter add** command. See Section 6.7.1 on page 137 for a description of each argument. Note that only the *<filnum>* and **report|r|hide|h** arguments are required whereas the remaining arguments are optional. This lets you easily change an existing filter from a report filter to a hide filter, without needing to completely redefine the filter.

### 6.7.6 Changing a SAP Report Filter

Use the **sap-report-filter chng** to change the definition of an existing SAP accept filter. Here is the syntax for this command:

```
sap-report-filter|srf chng|c
        <filnum> report|r|hide|h|hide-nearest|h
        [<svr-type> <svr-name>|* <snd-if-nw>|all]
```

The arguments for this command are the same as the arguments for the **sap-report-filter add** command. See Section 6.7.2 on page 138 for a description of each argument. Note that only the *<filnum>* and **report|r|hide|h|hide-nearest|h** arguments are required whereas the remaining arguments are optional.

### 6.7.7 Deleting a RIP Report Filter

Use the **rip-report-filter del** command to delete the definition from a filter number. Here is the syntax for this command:

```
rip-report-filter|rrf del|d <filter-list>|all
```

where:

*<filter-list>*|**all** Specifies the filter(s) you want to delete. You can specify a single filter number, a comma-separated list of filter numbers, or **all** for all filters.

### 6.7.8 Deleting a SAP Report Filter

Use the **sap-report-filter del** command to delete a SAP report filter definition. Here is the syntax for this command:

```
sap-report-filter|srf del|d <filter-list>|all
```

where:

*<filter-list>*|**all** Specifies the filter(s) you want to delete. You can specify a single filter number, a comma-separated list of filter numbers, or **all** for all filter definitions.

# Index